Optimised Design and Analysis of All-Optical Networks

Glenstrup, Arne John

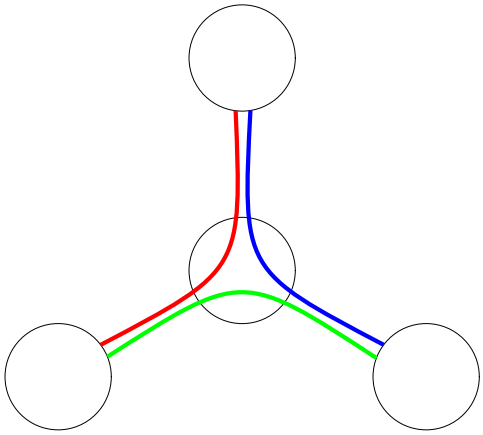
Publication date:
2002

Document Version
Publisher's PDF, also known as Version of record

Optimised Design and Analysis of All-Optical Networks

Arne John Glenstrup

PhD thesis



Research Center COM

Tele Danmark

Arne John Glenstrup
Optimised Design and Analysis of All-Optical Networks
Dansk titel: **Optimeret Design og Analyse af Fuldt Optiske Netværk**
PhD Thesis, Summer 2002

Research Center COM
Technical University of Denmark
DK–2800 Kongens Lyngby
Denmark

Tele Danmark
Teglholmsgade 1
DK–0900 Copenhagen C
Denmark

This thesis is electronically available at http://www.com.dtu.dk/staff/ajg/, http://www.tele.dtu.dk/~panic/ or http://www.diku.dk/~panic/

This document has been typeset by the author using the LaTeX system with the following packages: amsfonts, amssymb, anysize, babel, fontenc, graphicx, inputenc, makeidx, mathligs, mathpple, natbib, rotating, supertabular, varioref, xy

The main text is set in 10pt Palatino Roman. Sans serif and `typewriter` fonts as well as math $\mathcal{CALLIGRAPHIC}$ fonts are the standard Computer Modern fonts of the TeX system, and $\mathbb{BLACKBOARD\ BOLD}$ fonts are from the AMSFonts distribution. Small text is set at 9pt, while footnotes are set at 8pt.

Abstract

This PhD thesis presents a suite of methods for optimising design and for analysing blocking probabilities of all-optical networks. It thus contributes methodical knowledge to the field of computer assisted planning of optical networks.

A *two-stage greenfield optical network design optimiser* is developed, based on shortest-path algorithms and a comparatively new metaheuristic called *simulated allocation*. It is able to handle design of all-optical mesh networks with optical cross-connects, considers duct as well as fibre and node costs, and can also design protected networks. The method is assessed through various experiments and is shown to produce good results and to be able to scale up to networks of realistic sizes.

A novel method, *subpath wavelength grouping*, for routing connections in a multigranular all-optical network where several wavelengths can be grouped and switched at band and fibre level is presented. The method uses an unorthodox routing strategy focusing on common subpaths rather than individual connections, and strives to minimise switch port count as well as fibre usage. It is shown to produce cheaper network designs than previous methods when fibre costs are comparatively high.

A new optical network concept, the *synchronous optical hierarchy*, is proposed, in which wavelengths are subdivided into timeslots to match the traffic granularity. Various theoretical properties of this concept are investigated and compared in simulation studies.

An *integer linear programming model for optical ring network design* is presented. Manually designed real world ring networks are studied and it is found that the model can lead to cheaper network design. Moreover, ring and mesh network architectures are compared using real world costs, and it is found that optical cross-connects should be drastically cheaper if they are to compete on the cost with ring networks.

An *MPλS (multi-protocol wavelength switching) simulator* is constructed, focusing especially on the timing in the setup phase to asess the blocking probability effect of node and link delays as well as wavelength reservation strategies. Initial results suggest that node delays have a greater effect than link delays, and that only few wavelengths should be reserved during path setup.

Finally, a *novel way of calculating blocking probabilities* in all-optical WDM (wavelength division multiplex) networks by re-using existing blocking probability calculation tools is presented. The WDM network problem is transformed into a graph colouring problem whereupon information required by existing tools is extracted.

Resumé

Denne ph.d.-afhandling præsenterer en række metoder til optimeret design af fuldt optiske netværk samt til analyse af deres spærringsegenskaber, og bidrager således med metodisk viden til det område der handler om computerstøttet optisk netværksdesign.

Et *tofaset program til optimering af greenfield design af optiske netværk* udvikles, baseret på korteste-veje-algoritmer og en forholdsvis ny metaheuristik, *simuleret allokering*. Programmet er i stand til at håndtere design af fuldt optiske maskenetværk med optiske cross-connects, inddrager såvel tracé- som fiber- og knudeudstyrsomkostninger, og kan også designe beskyttede netværk. Metoden vurderes gennem diverse forsøg og viser sig at producere gode resultater og kunne skaleres op til netværk af realistiske størrelser.

En nyskabende metode, *subpath wavelength grouping*, til at rute forbindelser i et multigranulært fuldt optisk netværk hvor bølgelængder kan bundtes og switches på bånd- og fiberniveau præsenteres. Metoden anvender en uortodoks rutningsstrategi der fokuserer på fælles delveje fremfor individuelle forbindelser, og søger at minimere såvel antallet af switchporte som fiberforbruget. Den viser sig at producere billigere netværksdesign end tidligere metoder når fiberprisen er relativt høj i forhold til knudeprisen.

Et nyt optisk netværkskoncept, det *synkrone optiske hierarki*, foreslås hvori bølgelængder underopdeles i timeslots for at kunne svare til trafikgranulariteten. Forskellige teoretiske egenskaber ved konceptet undersøges og sammenlignes i simuleringsstudier.

En *linær/heltalsprogrammeringsmodel for optisk ringarkitekturdesign* præsenteres. Manuelt designede ringnetværk fra den virkelige verden studeres og det viser sig at modellen kan føre til billigere netværksdesign. Desuden sammenlignes ring- og maskenetværksarkitekturer under anvendelse af priser fra den virkelige verden og det viser sig at prisen på optiske cross-connects skal reduceres drastisk hvis de skal kunne konkurrere på prisen med ringnetværk.

En *MPλS (multi-protokol bølgelængde-switchet) simulator* konstrueres med fokus på timingen i MPλS-opsætningsfasen for at vurdere spærringseffekten af såvel knude- og kantforsinkelser som bølgelængdereserveringsstrategier. De første resultater tyder på at knudeforsinkelser har større indvirkning end kantforsinkelser, og at kun få bølgelængder bør reserveres under MPλS-opsætningsfasen.

Sluttelig præsenteres en *nyskabende metode til at beregne spærringssandsynligheder* i fuldt optiske WDM-netværk (bølgelængdemultiplexningsnetværk) ved at genanvende eksisterende spærringsberegningsværktøjer. WDM-netværksproblemet transformeres til et graffarvningsproblem hvorefter den information som eksisterende værktøjer skal bruge uddrages.

Preface

This thesis is submitted in partial fulfillment of the requirements for the PhD degree at the Research Center COM of the Technical University of Denmark. The thesis reports work done in the period 1st June 1999–31st May 2002, under the supervision of Villy Bæk Iversen, Research Center COM, Bjarke Skjoldstrup, TDC Tele Denmark, and Lars Dittmann, Research Center COM. This industrial PhD study number EF794 is a cooperation between TDC Tele Danmark, Research Center COM and the Danish Academy of Technical Sciences, funded by Tele Danmark and the Danish Agency for Trade and Industry.

The PhD study is one of three in the *explain project*, a joint project between Research Center COM, the Department of Informatics and Mathematical Modelling (Technical University of Denmark), TDC Tele Danmark and Ericsson. As this PhD study is part of the explain project it is natural that the work presented here to a greater or lesser extent has been performed jointly with the other PhD students of the project, Christian Fenger and Thomas Stidsen.

The thesis concerns various teletraffic problems in backbone optical networks, and is organised into four parts: Introduction, Static Teletraffic, Dynamic Teletraffic and Conclusion. Additional work not central to the main thesis, as well as elaborations on results are given in Appendices A through C. The author's main contributions with this thesis are in Chapter 2 and Section 3.3 which report on the explain project, as well as Section 3.1, reporting on research performed during a visit to NTT Network Innovation Laboratories, Japan, in the spring of 2001, and Section 5 which is inspired by previous work (Brustenga, 1999; Glenstrup and Iversen, 2001).

The origins of the chapters are as follows: Chapter 2 is a major reworking and extension of work reported by Pióro et al. (2000) and Glenstrup et al. (2000). In Chapter 3, Section 3.1 reports on joint work with Naohide Nagatsu and colleagues at NTT Network Innovation Lab. Section 3.2 is an article written jointly with Christian Fenger (Fenger and Glenstrup, 2002); Section 3.3 is written jointly with Thomas Stidsen and Christian Fenger; and Section 3.4 is a reprint of an arti-

cle written jointly with Thomas Stidsen (Stidsen and Glenstrup, 2002). Chapter 4 is written solely by the author, as well as Chapter 5. Finally, Appendix C is an update of an article written jointly with Villy Bæk Iversen (Iversen et al., 2000).

Acknowledgements

I would like to thank all the individuals and organisations that have helped me during my PhD studies and the writing of this thesis.

My supervisors, Villy Bæk Iversen, Bjarke Skjoldstrup and Lars Dittmann have always been willing to help me out at times when I felt I was stuck, and even more frequent interaction with my colleagues Christian Fenger and Thomas Stidsen has led me to deeper insights—and hopefully a better thesis.

During my visit to North Carolina State University I had several pleasant and inspiring discussions with Harry Perros and George Rouskas, as well as help in practical matters.

I felt very welcome at the NTT Network Innovation Lab, and enjoyed numerous social events with everyone there, including an unforgettable hanami with sushi and sake. Special thanks go to Naohide Nagatsu for pointing me to wavelength grouping, for believing in me during long weeks of endless programming and for assisting me in the many practical matters involved when visiting Japan.

Thanks go also to Michał Pióro for teaching me and my colleagues about network optimisation and having the patience to coauthor an article with us.

Henning Christiansen has been a tremendous help whenever I had problems with the computing systems. His very efficient help at practically any time of day have been quite invaluable to me.

I would also like to thank the people of the Network Group at Research Center COM for the very friendly and open atmosphere that always makes me feel welcome, and similar thanks go to my colleagues at TDC Tele Danmark; especially Helle Glarø Mathiesen has been a great help.

I would like to express a big thanks to Jill Glenstrup for proofreading large parts of the manuscript in the draft stage. She found many typos and errors, but any that remain are still my responsibility!

Finally, I am indepted to my friends and the people I live together with, especially Kristine Hommel, for their kindness and patience with me especially in the last hectic phase of the writing process in which I have been far less sociable than I should be.

Summer 2002

Arne John Glenstrup

Contents

Part I

Introduction

Chapter 1

Introduction

It will come as no surprise that the need for telecommunication is continuously increasing, and while access technologies are becoming increasingly mobile and intelligent (e.g., wireless local area networks, global positioning systems, mobile agents), the need for broadband backbone data transport persists, be it by radio based or optical fibre links.

This thesis is about planning and analysing the optical networks that constitute the main vehicle for the backbone networks:

- Ways of optimising the layout of networks of various models: where to deploy how many fibres and switches
- Analysing the quality of a network once it is deployed

Suppose a telecommunication operator like TDC Tele Danmark decides to deploy a backbone network in, say, India as a basis for entering the Indian market for telecommunication services—this is an example of a *greenfield network design scenario*. One of the first tasks would be to choose where to locate the nodes of the backbone, and it would be sensible to place them in major cities where the larger communication demands originate. One choice of network nodes is shown in Figure 1.1 on the facing page.

Choosing such a fairly small set of nodes right might not seem easy, and should be based on knowledge of expected communication demands, but the problems do not stop here. The nodes must be interconnected in such a way that the expected bandwidth requirements can be satisfied, subject to various technical, political and economic constraints. Even ignoring the technical and political

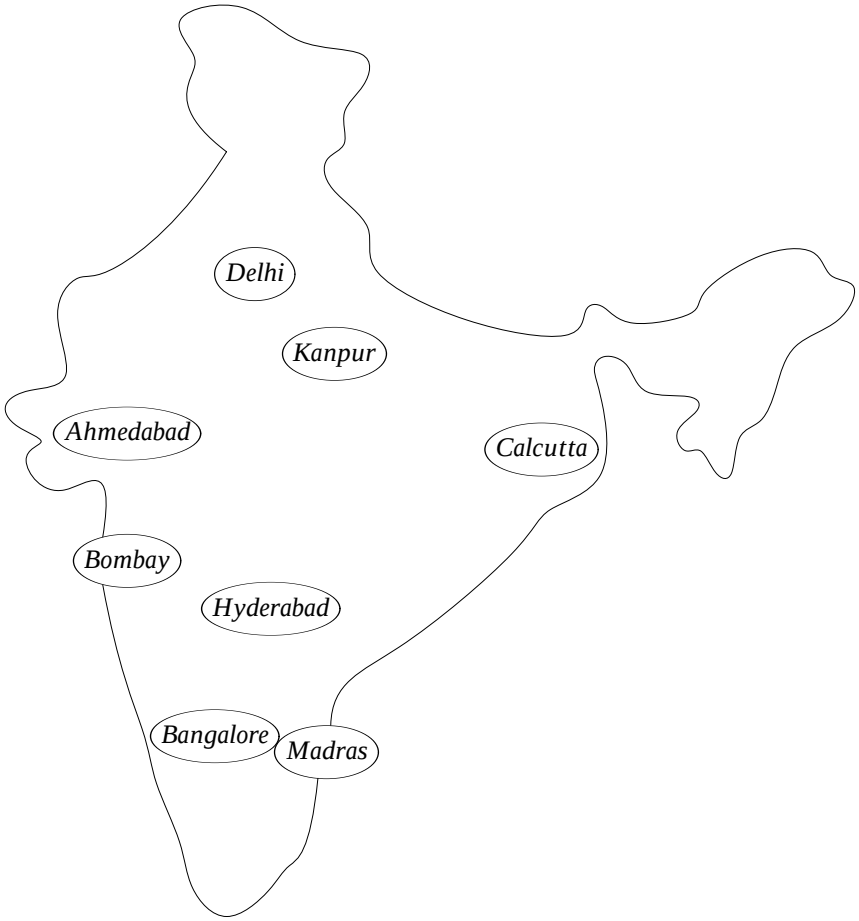


Figure 1.1: Example backbone network nodes in an Indian network

constraints and just focusing on constructing the cheapest network is nontrivial: Some examples of how to interconnect the example nodes producing star, ring, multi-ring and mesh networks are shown in Figure 1.2. The various solutions

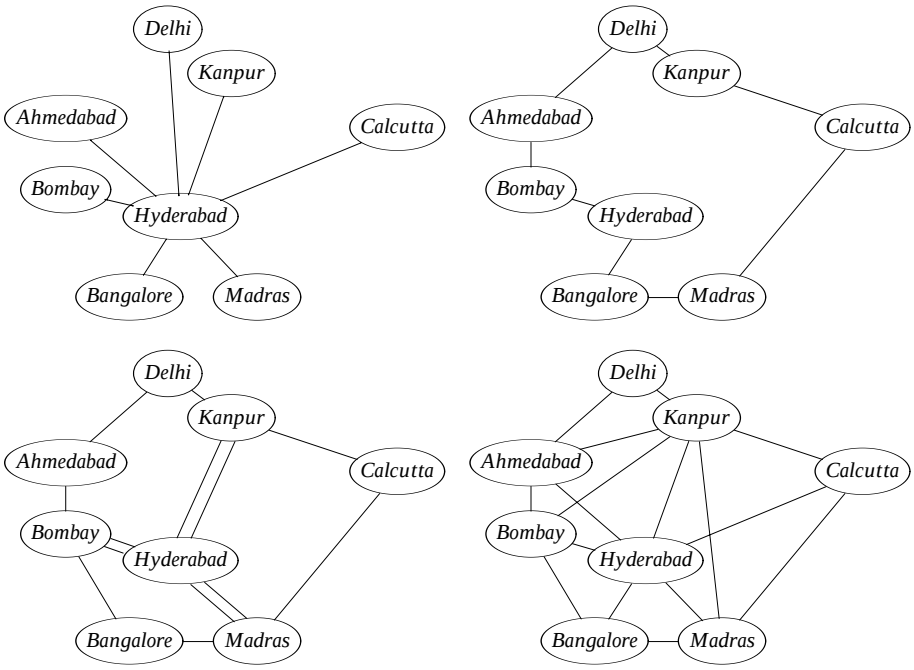


Figure 1.2: Examples of interconnecting the nodes of Figure 1.1

have different merits: star networks make the routing of connections simple; ring networks are the simplest way to ensure that any pair of nodes is connected by at least two different routes; multi-ring networks provide shorter paths and thus lower fibre requirements; and mesh networks generally lead to the lowest fibre requirements. Even if the network planner chooses just one of these general architectures, there are still overwhelmingly many choices of exactly where to locate the links, and choosing correctly is important, for there can be great variations in the cost of deploying a network, according to which links are selected. Furthermore, the best choice depends not only on the distances between nodes but also on the communication traffic pattern.

A network is designed on the basis of expected traffic demands, but once it has been built, the actual traffic it must carry may differ from the original expec-

tations. This is particularly true for networks that are used for *dynamic* traffic, similar to the public switched telephone network where connections are established on demand instead of reserving a fixed bandwidth for each customer. The quality of a network carrying dynamic traffic can be assessed by measuring the risk of not being able to establish a connection—the *blocking probability*. It is important for an operator to know whether the blocking probabilities are within prescribed limits, and whether they are approaching these limits as this could be an indication of poor resource usage or the need for more network capacity.

The objective of this thesis is thus to contribute to the research in how network planners can be assisted by computer programs for optimising the layout and cost of optical networks, as well as analysing their performance.

The goal of the thesis is to design and evaluate concrete software tools for network planning and analysis.

1.1 Thesis overview

Throughout the thesis we assume the reader has basic knowledge of optical networking and teletraffic concepts covered by standard textbooks (Iversen, 1999; Ramaswami and Sivarajan, 1998; Sexton and Reid, 1997).

In Part II of the thesis we shall consider static teletraffic, where the nodes and traffic demands are given and the objective is to design a network in a greenfield scenario, satisfying the demands at a minimised cost:

In Chapter 2, considering just the link costs, we present mathematical definitions for a list of network design problems, followed by a description and evaluation of a tool we have developed for solving these problems, the *explain tool*.

Chapter 3 concerns various design aspects where the node costs are also included: Section 3.1 describes an algorithm for minimising the cost of optical networks where the nodes are equipped with multigranular switches that are able to switch connections at wavelength as well as band (a group of wavelengths) and fibre level; Section 3.2 considers the opposite problem, presenting a way of dividing the capacities of wavelengths in optical fibres into smaller units using time division multiplexing, thus making them more suitable for present day traffic demand sizes; Section 3.3 is an extension of the ideas presented in Chapter 2 to include node costs as an integrated part of the optimisation process; and Section 3.4 presents a method for using general mathematical optimisation tools for designing ring networks.

In Part III of the thesis we shall consider dynamic teletraffic, presenting two ways of analysing optical networks:

In Section 4 we consider specifically the effect of *wavelength reservations* in MPλS (multi-protocol wavelength switching) networks and construct a simulator for determining the effect that the node processing and link transmission delays experienced by control messages have on the blocking probabilities.

In Section 5 we present a method for converting the additional constraints offered by optical networks into a form similar to the constraints of conventional circuit-switched networks, thus facilitating re-use conventional tools for calculating blocking probabilities.

The thesis is concluded in Part IV with a list of topics for future research.

The remaining part of the present chapter gives a brief overview of some basic concepts used throughout the thesis and reviews relevant literature.

1.2 Optical networks

Backbone telecommunication networks are characterised by having relatively few nodes (typically less than 50) but supporting large communication channel bandwidths. Such large bandwidths are transported most cost-efficiently on optical fibres, which have been deployed in telecommunication networks during the past couple of decades. At first, simple point-to-point fibre links using just one wavelength were deployed; later, they were upgraded to exploit the fibre optical capacity by attaching wavelength division multiplexing (WDM) equipment at the ends. This equipment consists of tunable or fixed-wavelength transmitters / receivers and (de)multiplexers that enable simultaneous communication in the same fibre using different wavelengths. This helps satisfy the exponential increase in communication demand, and optical fibres can provide reliable communication over long distances with low delay. While delays are low in fibres, this is not so in the conventional digital cross-connect (DXC): When the path from source to destination node involves several hops, the optical signal is at each DXC converted into electrical form before it is switched and converted back into an optical signal for the next hop. This optical-electrical-optical (OEO) conversion is a delay bottleneck in current optical backbone networks, which has spurred the development of the all-optical network (AON).

1.2.1 The all-optical network

The AON utilises the low delay in optical fibres by keeping the signal in the optical domain from source to destination. Each node of the AON is equipped with an optical cross-connect (OXC) or an optical add / drop multiplexer (OADM), both of

which are able to pass on the optical signals without OEO conversion, thus eliminating electrical delay. A good overview of WDM and AON can be found in the literature (Sengputa and Ramamurthy, 2001).

1.2.2 WDM

When WDM is used in the AON we obtain the *wavelength routed optical network* (WRON). WRONs transmit signals entirely in the optical domain, reducing delay, but imposing a new restriction, the *wavelength continuity constraint:* When routing a connection, all links along the route must use the same colour. The connection is thus assigned a *lightpath*, identified by its route and wavelength. The wavelength continuity constraint restricts the number of multihop paths that can be accommodated, because two different demands cannot use lightpaths that share a fibre if they have the same wavelength. This gives rise to the *routing and wavelength assignment* (RWA) problem, where all traffic demands must be assigned lightpaths.

Consider the simple 3-link star network in Figure 1.3, where each link has a capacity of 2 wavelengths (indicated by two lines), and suppose three connections are to be made: along links *AB*, links *AC* and links *BC*. In an electrically switched network, they can simply be routed along the given links. However, in an all-

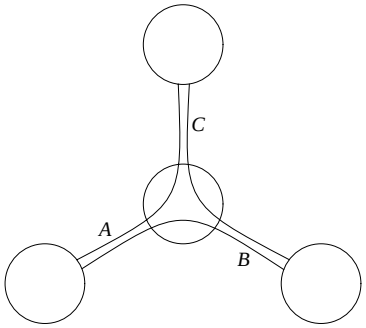


Figure 1.3: Routing three calls *AB*, *AC* and *BC*.

optical WDM network with a capacity of two wavelengths on every link, say red and green, it is not possible to assign wavelengths to the three connections of the example because each connection must (in this example) be assigned a different wavelength than either of the others.

Solving the RWA problem is in general an NP-complete problem[1] (Garey and

[1] cf. Section 1.4.5 on page 20

Johnson, 1979; Chlamtac et al., 1992), but Ramaswami and Sivarajan (1995) devised an linear programming (LP) program to calculate an upper bound on the amount of carried traffic that *any* RWA algorithm would be able to accommodate. Various theoretical resource bounds for WDM networks are given by Aggarwal et al. (1996), and a good survey of the RWA problem is presented by Zang et al. (2000).

The wavelength continuity constraint can be removed by equipping the nodes with wavelength converters, which allows the optical signal to change wavelength from the incoming to the outgoing link—without OEO conversion. However, as this makes the nodes more complex and expensive, much research has investigated how large the benefits are. Most results indicate that the advantage of wavelength converters is negligible when designing WRONs without backup paths, but Nagatsu et al. (1996) found that when considering backup paths, significantly more wavelengths are required, especially if the same wavelength must be used on primary and backup paths. Garnot and Masetti (1997) reached similar results, and Baroni et al. (1999) found wavelength conversion is beneficial particularly when designing networks with high connectivity, and networks with additional capacity for future traffic growth. It seems that the best strategy for obtaining a high utilisation is to place wavelength converters at few, selected nodes (Belotti and Stidsen, 2001; Garnot et al., 1997), also for dynamic traffic (Späth and Bodamer, 1999).

Note that in this thesis we shall only consider components of the optical network—fibres, OXCs and OADMs—at an abstract level: Fibres contain a number of wavelengths and interconnect OXCs / OADMs. We gloss over details of optical signal regeneration, dispersion and other physical details.

1.2.3 Network control and management

The software and hardware systems that supervise the networks are conceptually divided into *control* and *management planes*.

The control plane takes care of operations on a small time scale: performing admission control for connection demands, setting up and tearing down short term on-demand connections, responding to network element errors and switching to backup capacity (and in some cases determining which backup capacity to use).

In contrast, the management plane takes care of operations on a longer time scale: configuration, fault, performance, accounting and security management. This includes setting up policies for admission control and priorities, reserving and assigning capacity for long-term connections, determining where backup capacity should be located (and in some cases the mapping between primary and

backup capacity), and reporting network errors to the network operator.

An example are the multiplex section shared protection ring (MS-SPRing) networks we consider in Chapter 3.4: the connections are routed and the OADMs are set up by the network management plane, while link failures are detected by the control plane which also takes care of switching broken connections to backup fibres.

The control plane is constituted of signalling protocols between network elements at a low level, whereas the management plane handles signals at a higher, aggregated level. In practice this means that management software must communicate with network elements from different vendors, which can cause problems due to conflicts between proprietary protocols, especially in areas where standardisation has not been fully designed or implemented.

1.3 Teletraffic

While fibres and switches are the optical components that supply the data transport capacity, they are only one part of network planning and design problems; *teletraffic* is what makes up the other part. Teletraffic covers all demands for communication bandwidth, be they for voice telephony or data transferral; however, in the backbone network all traffic is simply considered as data traffic because any narrowband traffic like telephone connections are aggregated into broadband data connections.

1.3.1 Static and dynamic traffic demands

Designing an AON usually means that traffic demands—i.e., requests for transferring data between pairs of nodes—must be satisfied by specifying how much capacity in the form of fibres and switches must be used in the AON links and nodes. If there are given limits on the number of fibres on each link or the switch size in each node, the network design problem is said to be *capacitated,* otherwise it is *uncapacitated.*

Research into AON design has considered dynamic traffic demands (Chlamtac et al., 1992; Mohan et al., 2001; Zhang and Acampora, 1995; Zang et al., 2000) in which connection requests arrive and are routed one at a time, as well as static traffic demands (Baroni et al., 1999; Chlamtac et al., 1992; Cwilich et al., 1999; Doshi et al., 1999; Garnot and Perrier, 1998; Kershenbaum et al., 1991; Lee et al., 1998a; Nagatsu et al., 1996; Saniee, 1996; Sinclair, 1999; Varela and Sinclair, 1999) in which the demands are given as a fixed set. Dynamic traffic demands may experience *blocking:* if there are not sufficient network resources to accommodate a request when it arrives, it is blocked.

1.3.2 Blocking probabilities for dynamic traffic

Blocking probabilities exist in a scenario where the capacities in a network have been decided and dynamic traffic demands, known as *calls*, arrive.

Call arrivals can be modeled as stochastic point processes, typically one for each node pair in the network. Such an arrival process can be characterised by the distribution of the interarrival time; that is, the time between two successive arrivals. An arrival process is *stationary* if it does not depend on the absolute time at which it is observed; it is *independent* when future arrivals depend only on the state the process is in and not how this state was reached; and it is *regular* if two arrivals never occur at exactly the same time.

The most basic arrival process used to characterise arrivals in telecommunication networks is the Poisson process which is stationary, independent and regular. The interarrival times t of the Poisson process follow a negative exponential distribution $F(t) = 1 - e^{-\lambda t}$, where λ is the *arrival intensity* indicating the average number of arrivals per time unit. The Poisson process is best for modelling call arrivals from a large set of sources that are not expected to be correlated (e.g., telephone subscribers). A multitude of other distributions have been devised for modelling other scenarios like overflow traffic, traffic from a limited set of sources etc.

We will associate each call with a *holding time*, which is the time that the call will occupy the connection when it is served. As for the interarrival times of the arrival process, holding times are also characterised by a distribution.

The *offered traffic*, A, of a traffic demand is the amount of data transmission performed if no calls are blocked. For a Poisson process with intensity λ and mean holding time $1/\mu$ the offered traffic is λ/μ. When a call arrives the system may decide (e.g., because there is no unused capacity left) to *block*—i.e., not serve—the call. The *carried traffic*, Y, of a traffic demand is the amount of traffic which is not blocked.

A system serving calls can be modelled using the *lost calls held* model, where blocked calls are put on hold and served when the system again is able to do so. However, in this thesis we shall only consider the *lost calls cleared* model, where any blocked calls disappear and do not affect the future behaviour of the system.

The *call blocking* is the probability that a call is blocked, while the *time blocking* is the fraction of the time that any arriving calls will be blocked (whether or not any arrive in this time), and the *traffic blocking* is the fraction of traffic which is blocked (i.e., $(A - Y)/A$). For the Poisson process all three blocking probabilities are identical.

1.3.3 Circuit switching and packet switching

When data is to be transported from a source node to a destination node that are not connected by a direct link, the network must make several intermediate links and nodes, known collectively as a *path* or *route*, work together in tandem to supply the requested bandwidth. There is a spectrum of ways between two extremes, *circuit switching* and *packet switching*, by which this can be accomplished.

In circuit switched networks a route is selected and the intermediate nodes set up before the data is transmitted, and subsequently torn down when the connection is no longer needed. Conversely, in packet switched networks data is forwarded hop-by-hop, where each node examines the packet header to determine the next node to which it must be forwarded. This can change due to network failures and congestion, causing packets between the same source-destination node pair to use different paths which can increase the risk of receiving packets in the wrong order. However, packet switching is generally more robust against failures than circuit switching, as packets can dynamically be routed around the failed elements.

An asynchronous transfer mode (ATM) network is an example of a connection oriented packet switched network, in which *virtual channels* are created by setting up the switching tables in the intermediate nodes. Data is then transferred in cells, and each node need only perform very simple lookups in its switching table to determine next hops (Sato, 1996; Sexton and Reid, 1997).

Traditional Internet protocol (IP) networking is an example of connectionless packet switching where each IP packet contains the destination address, so that intermediate nodes must perform nontrivial address lookup—usually based on matching a prefix of the address—to decide the direction in which the packet must be sent.

1.3.4 MPLS, MPλS and GMPLS

When the Internet really took off in the 1990's the increasing demand for Internet connections with delay and bandwidth guarantees spurred the development of protocols that could control and deliver such specified *quality of service* (Xiao and Ni, 1999). The first development, the *IntServ* model, required that all the intermediate router nodes (core routers) kept track of each individual connection that traversed them, to be able to guarantee the promised bandwidth and delay characteristics. The general opinion was that this fine-grained processing severely limits the speed of core routers, so instead development has continued on an alternative, the *DiffServ* model. Under this model routers accepting traffic to be sent across the network (edge routers) assign one of a limited set of forwarding equivalence classes to each connection. Connections in the same class arriving at the

same node receive the same quality of service, but without a definite guarantee—this way core routers need only make a coarse-grained distinction of connections based on these classes.

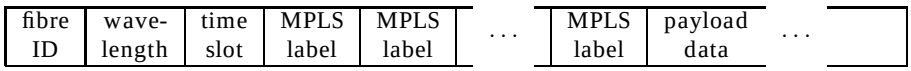
Originally the Internet was connectionless and all packets were routed hop-by-hop, based on the destination address specified in the packet header. As the Internet address space got filled up, the process of looking up addresses—by matching longest prefixes—put more load on core routers, and multiprotocol label switching (MPLS) was seen as a good solution to this problem. A router forwarding MPLS packets need only make lookups based on total (i.e., not prefix) matches of a short label which is only significant within each link interface. This speeds up MPLS core routers, but advances in gigabit router technology has reduced this advantage significantly, and the main advantage of MPLS is now that it enables forwarding of packets along specified, non-shortest paths (Armitage, 2000). This allows for *traffic engineering*, whereby traffic can be efficiently routed along paths with specific quality of service.

Packets are sent across an MPLS network along label switched paths (LSPs), which are represented by the input-output label / interface mappings in each core router. It is important to notice that MPLS in itself only replaces the forwarding mechanism; routers must still run routing calculation and network state information protocols to obtain the information used to set up LSPs. Thus network planning of IP networks running MPLS is very similar to traditional IP networks—the main differences being how to utilise traffic engineering and how to provision sufficient label space on each link (Lawrence, 2001).

The link local significance of labels is probably *the* reason for the success of MPLS, and observing that local significance is true also of wavelengths on a WDM link leads to the realisation of MPλS . In MPλS optical switch setup corresponds to setting up MPLS input-output label mappings, and lightpaths correspond to LSPs. The main differences lie in granularity and label identification (Awduche and Rekhter, 2001). Regarding the former, the number of wavelengths on a WDM link is an order of magnitude smaller than the typical MPLS link label space. For the latter, the label of an MPλS packet is implicit in its transmission wavelength, while the label of an MPLS packet is part of the packet header. This allows MPLS routers to *push* labels; that is, to add a new label to the front of the packet before forwarding it. Pushing labels enables aggregation of several LSPs for tunnelling across routers and thus reducing the label space requirements on tunnel core routers.

The extension of MPLS does not stop here: the Internet Engineering Task Force have been developing generalised multiprotocol label switching (GMPLS), in which the MPLS principles are used in the control plane at several network levels: packet, time slot, wavelength and fibre level (Banerjee et al., 2001a,b). Ob-

viously, it is not possible to push an MPλS wavelength or a timeslot onto another, but instead pushing can be interpreted as aggregating several LSPs into one higher level LSP, thus forming a hierarchy of GMPLS networks. The label stack on each data packet represents this hierarchy:



A key constraint here is that GMPLS LSPs always start and terminate on switches of the same architecture because the terminating node must be able to handle the header of the received packet.

1.4 Network Design

Network design is the art of combining network infrastructure and operation strategies to satisfy certain traffic demands. In this section we will briefly review central concepts concerning design goals, types, methods and tools.

1.4.1 Network design goals

When considering dynamic traffic, arrival intensities are given for each node pair demand, and the design goal is to determine the capacities of the network links and nodes, minimising the blocking probabilities. In most cases, if it is not given as input, the *routing* of each connection is also a result of the design process. Alternatively, Jan et al. (1993) present an algorithm for maximising "sys-reliability," which is a measure of the probability that two nodes are able to communicate with each other, given a certain reliability for each link.

The design goals for a network with static traffic demands in a capacitated setting include maximisation of the total network flow (Saniee, 1996) or number of accepted demands (Chlamtac et al., 1992) or minimisation of network congestion (Dutta and Rouskas, 1999). In the uncapacitated setting, the design goals include minimisation of the maximum number of required wavelengths on any link (Baroni et al., 1999; Chlamtac et al., 1992; Varela and Sinclair, 1999), minimisation of total network cost (Doshi et al., 1999; Kershenbaum et al., 1991; Lee et al., 1998a; Sinclair, 1999), minimisation of the number of hops (Garnot and Masetti, 1997) or maximisation of wavelength utilisation (Nagatsu et al., 1996).

1.4.2 Greenfield network design

Even in the uncapacitated scenario it can be natural to speak of a predeployed capacity on each link and node. This capacity then has zero cost, and the task is

the *network extension design* problem, where *additional* capacity must be deployed. In contrast, the *greenfield network design* starts from scratch: there is no "capacity for free" at the outset.

Link costs are often taken to be proportional to the number of deployed fibres, but for the greenfield model where also ducts must be constructed, it can make good sense to include an opening cost for each link (Pióro et al., 2000; Glenstrup et al., 2000), or simply that it is an increasing, concave function of their length (Kershenbaum et al., 1991).

Finally, if all link costs are set to 1, the resulting design will tend to comprise paths with few hops—at the cost of longer link lengths—which can be appropriate if node costs dominate (Garnot and Perrier, 1998; Hjelme and Andersen, 1999).

1.4.3 Multilevel network design

Optical networks can conceptually be viewed at several levels, usually at least three: the *optical*, *virtual* and *logical* levels as shown in Figure 1.4. The nodes

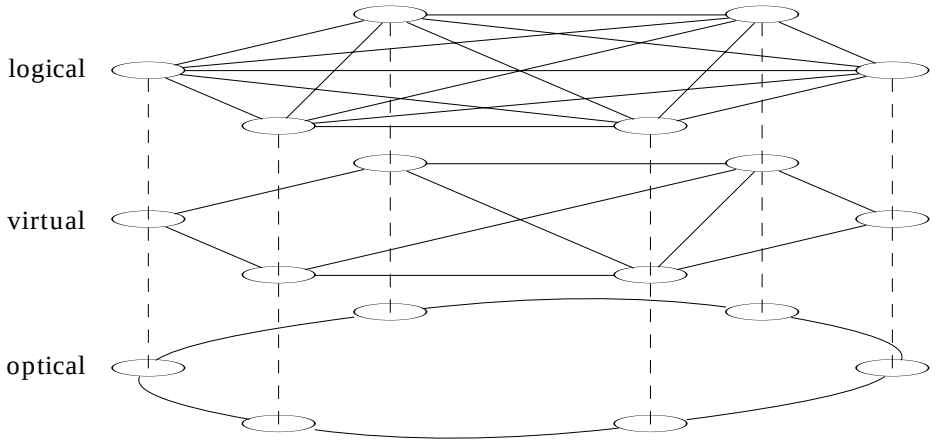


Figure 1.4: Viewing an optical network at three conceptual levels

are the same at all levels, while the links of the upper levels are realised by a path composed of one or more links in the level immediately below. The optical level consists of the physical components—fibres and optical switches—while the virtual level is a collection of virtual connections that have been provisioned to transport aggregated traffic flows from the logical level. The logical level is represented by the traffic demand matrix and is thus normally fully connected, while

the optical level is much more sparse. Dutta and Rouskas (1999) present integer linear programming (ILP) programs for designing multi-level WDM networks which also take the wavelengths into account.

When constructing optimisation algorithms for protected network design it is important to notice that two link disjoint paths in intermediate levels (e.g., the virtual level) might in fact share a common optical link. Links in intermediate levels which have an optical link in common are in the same *shared risk link group,* and optimisation algorithms must either optimise each level taking these risk groups into account, or handle the entire hierarchy, operating explicitly with optical link failures (Pióro, 1999).

1.4.4 Protection and restoration

Given the importance of telecommunication networks today, operators are increasing their focus on reliable networks which continue to provide service even when some network elements fail, for example due to cable cuts or power blackouts.

Proactive measures to ensure reliable connections include *protection* by deploying spare capacity in the network, and also precomputing static backup paths (Baroni et al., 1999; Cinkler, 2002; Cwilich et al., 1999; Doshi et al., 1999; Garnot and Masetti, 1997; Huang and Copeland, 2001; Lee et al., 1998b; Nagatsu et al., 1996; Saniee, 1996). Protecting primary paths by dedicating backup capacity to each path is called $1+1$ *protection* or $1:1$ *protection,* depending on whether the signal is sent over the backup path during normal operation or not. When one backup path is shared between n primary paths we have $1:n$ *protection* so the general case of a network with n primary paths and m backup paths should be termed $m:n$ *protection,* cf. Figure 1.5 on the next page. Reactive measures include *restoration,* the act of finding a backup path when a primary path is broken (Shami and Sinclair, 1999).

Two major protection methods are distinguished by the extent of the backup routing, cf. Figure 1.6 on page 17: In *link protection,* all traffic of the broken link is routed around the failure, between the two end nodes of the link. *Path protection* reroutes the entire path of any path that is broken for the given failure; if the non-broken links of the primary path cannot be reused—e.g., due to the management systems—we call it *non-reusable primary capacity* (NRPC). Link protection has the advantage of requiring only local decisions to perform the restoration, while path protection requires a more complicated protection switching protocol, which also increases the protection switching time. On the downside, link protection generally uses more spare capacity than path protection, partly because link protection usually makes the paths at least one hop longer, and partly because path pro-

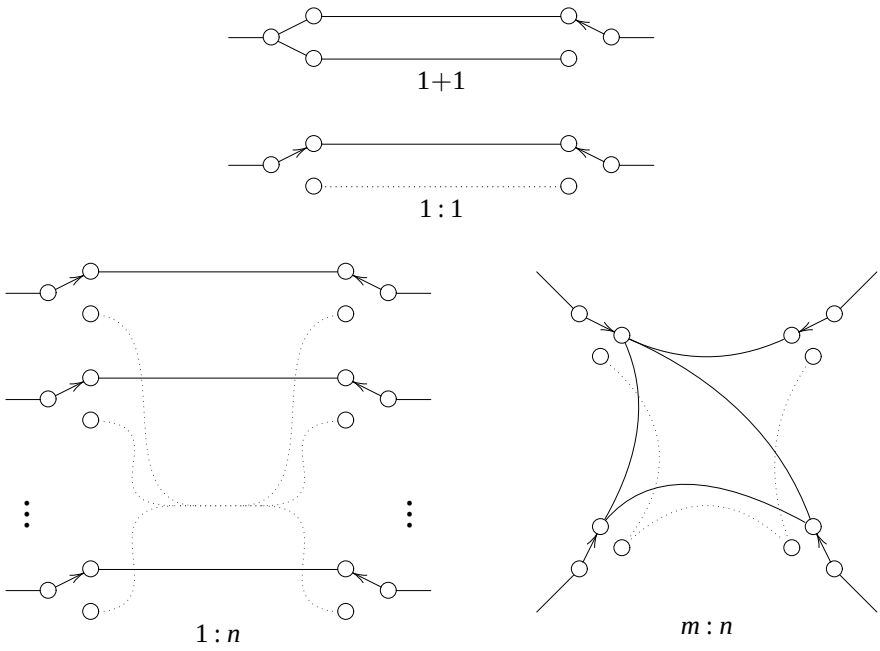


Figure 1.5: 1+1, 1 : 1, 1 : n and m : n protection. The switches at the end of a connection switch to the backup path—indicated by dotted lines—if the primary path fails

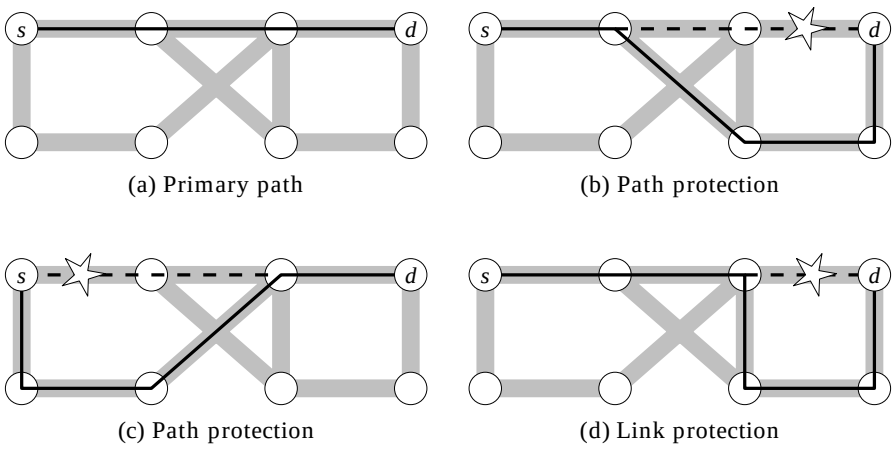


Figure 1.6: Link and path protection schemes

tection tends to “spread out” the rerouted lightpaths more in the spare network resources.

In recent years the ring network architecture called MS-SPRing (Sexton and Reid, 1997) has to some extent replaced the general mesh network architecture. MS-SPRing has become popular among network operators because it is able to switch quickly and automatically to backup paths based just on local decisions, that is, without the need for centralised failure control, cf. Figure 1.7 on the next page. The entire network is constructed from rings where each link in a ring has twice the maximal capacity needed on any of its links. When a link fails, a link protection scheme reroutes all traffic the long way round the ring. The major drawback with MS-SPRing is that it requires at least 100% spare capacity.

This rather severe underutilisation of link capacity led Grover and Stamatelakis (2000) to develop the *p*-cycle concept for protection of mesh networks. The principle is simply to make large cycles covering as many nodes as possible; then all connections between these nodes are protected, cf. Figure 1.8 on page 19. Protection switching using *p*-cycles can be as fast as for MS-SPRing networks, because if all the switches on the *p*-cycle are set up beforehand, just the end nodes of a failed link need to switch to a backup connection on the *p*-cycle. Furthermore, it can be proved that the required spare capacity is close to the minimum possible for any mesh network based link protection scheme (Stamatelakis and Grover, 2000). In practice Grover et al. got backup capacity requirements that were less than a 10% increase of that of mesh based link protection.

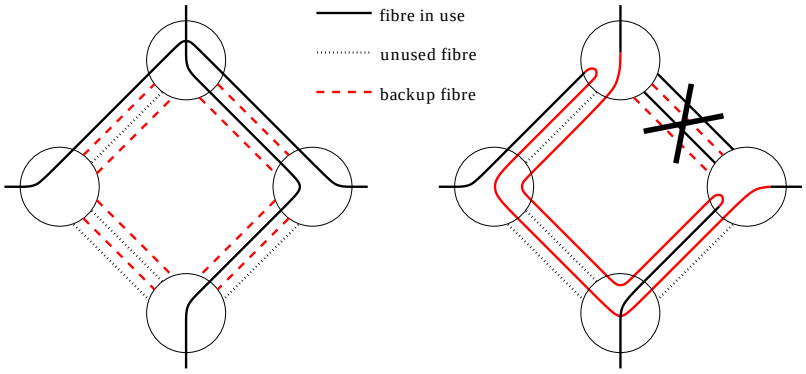


Figure 1.7: Ring protection: when a link failure occurs, its end nodes reroute the connections in the opposite direction along backup fibres

1.4.5 Network design methods

Optimal solutions to small network design problems (up to about 15 links) can be found by ILP tools (Baroni et al., 1999; Cwilich et al., 1999).

ILP programs consist of an objective $c_1x_1 + \cdots + c_nx_n$ and a list of constraints, all parameterised by the *decision variables* x_i which are free variables whose optimal values are to be determined. ILP constraints are written in the basic form $\sum_i a_i x_i \leq b$, but for presentation purposes we will also use forms like $\varphi = 1 \Rightarrow \sum_i a_i x_i \leq b$, where φ is a binary variable. These expressions can be rewritten into the basic form by straightforward rewriting rules, given that $M \neq 0$ is an upper bound and $m \neq 0$ a lower bound (possibly negative) on $\sum_i a_i x_i - b$, and $0 < \varepsilon \ll 1$:

Implication			*is implemented as*		
$\varphi = 1$	$\Rightarrow$	$\sum_i a_i x_i \leq b$	$\sum_i a_i x_i$	$\leq$	$b + (1-\varphi)M$
$\varphi = 1$	$\Leftarrow$	$\sum_i a_i x_i \leq b$	$\sum_i a_i x_i$	$\geq$	$b + \varphi(m-\varepsilon) + \varepsilon$
$\varphi = 1$	$\Rightarrow$	$\sum_i a_i x_i \geq b$	$\sum_i a_i x_i$	$\geq$	$b + (1-\varphi)m$

ILP solvers take as input an ILP program and produce as output an assignment of values to $x_1, \ldots, x_n$ that optimises the value of the objective function.

In general, the network design problems are variants of the *multicommodity flow problem,* which can be stated in an *arc-flow* formulation in which input and output flow is balanced for all nodes (Cinkler et al., 2000; Doshi et al., 1999; Dutta and Rouskas, 1999; Lee et al., 1998a), or a *link-path* formulation, where each traffic

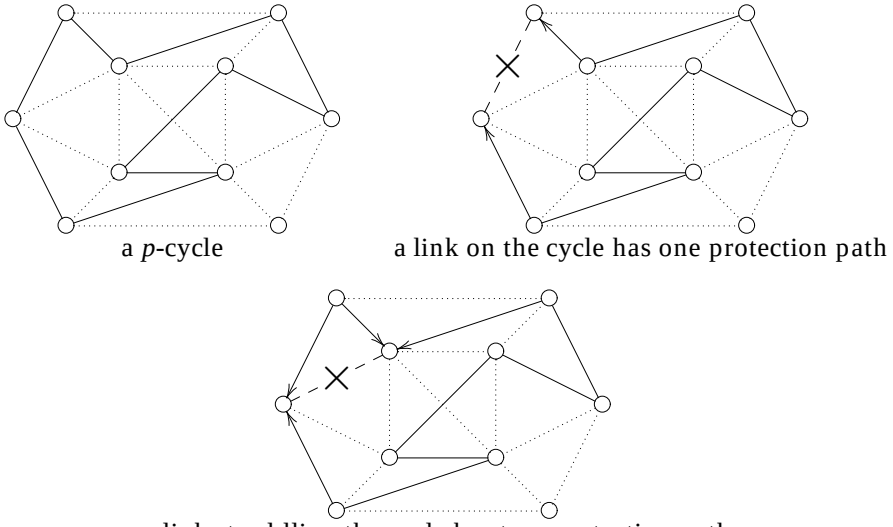


Figure 1.8: *p*-cycles can protect all links between its nodes (from Grover and Stamatelakis, 2000)

demand can be satisfied by a limited set of precomputed paths, which thus limit the optimality of the solution (Glenstrup et al., 2000; Pióro et al., 2000; Saniee, 1996), cf. Section 2.4.2 on page 42.

Doshi et al. (1999) give detailed algorithms for solving some ILP programs, based on Lagrangian relaxation and subgradient optimisation techniques, and Jan et al. (1993) develop a branch and bound algorithm. Planning just backup capacity, Lee et al. (1998b) used a branch and cut algorithm for finding the optimum in a 42 link network. Removing the integrality constraints and thus relaxing the problem to an LP program, Saniee (1996) was able to optimise the layout of a 150 link network with 29 switches. By designing a protection path finding algorithm based on local bypasses along primary paths, Cwilich et al. (1999) were able to find backup paths for a network of 1974 links and 1050 nodes satisfying 679 demands by running column-generation based ILP solvers for 60 hours.

However, for networks of realistic sizes, solving these network design problems optimally using ILP tools is not possible within reasonable computing time and memory because they are NP-hard problems, that is, no algorithm exists[2] that can find an optimal solution to any such problem of size n within time $p(n)$, where $p(n)$ is a polynomial. An NP-complete problem is an NP-hard problem for which a given solution can be verified in polynomial time (Jones, 1997). Thus either these network problems must be decomposed (Dutta and Rouskas, 1999), or reduced—for example by restricting the length of the paths for each demand (Baroni et al., 1999)—or *heuristics* must be applied.

Nagatsu et al. (1996) constructed a heuristic using shortest path algorithms with link weights set up to minimise the link usage variance, in an attempt to maximise utilisation. The wavelength continuity constraint was then handled by grouping non-conflicting paths together, and assigning a separate wavelength to each group. Baroni et al. (1999), Chlamtac et al. (1992) and Garnot and Masetti (1997) assigned lightpaths to demands in decreasing order of minimum hop distance, the idea being that assigning wavelengths to shorter paths is easier than longer paths. Strangely, though, Cinkler et al. (2000) reported good results based on the *opposite* sorting order, and furthermore that sorting according to demand volumes—satisfying larger demands first—improved the result.

Many heuristics also include a post-optimisation, where individual paths are reallocated to improve utilisation (Nagatsu et al., 1996; Doshi et al., 1999; Baroni et al., 1999).

Finally, various optimisation methods based on *stochastic algorithms* have been developed for network design problems (Sinclair, 1999; Varela and Sinclair, 1999).

[2]Strictly speaking no algorithm *has yet been found*—but although not proved, everybody believes it does not exist.

1.4.6 Network design tools

As network operators are beginning to see a decrease in the exponential growth in bandwidth demand, the manual and rule-of-thumb network deployment strategies of the past decade are being augmented with computer aided network planning. As the study in Section 3.4 indicates, manual planning does not a priori lead to the most cost-efficient network designs.

Several software packages for optical network design and analysis tasks are commercially available—examples of photonic network planning tools include COMPOSIS® (AixCom, 2002), LightPlan™ (Lightscape Networks, 2002), OPNET Modeler® (OPNET Technologies, Inc., 2002), StarNet Planner (Tellium, 2002) and VPItransportMaker (VPIsystems™, 2002). Additionally, many operators make use of in-house developed software tailored specifically to their choice of network architecture, and various ILP solvers exist for solving general network problems (Brooke et al., 1992; ILOG Cooperation, 2000).

Part II

Static Teletraffic

Chapter 2

Optimising networks for link costs

In this section we consider a greenfield scenario where a telecommunication operator has decided where to locate some WDM network nodes (e.g., major cities), and now wants to figure out how to interconnect the nodes with links at the least cost such that continuous service is guaranteed in the face of certain link or node failures. No architectural restrictions are imposed: The aim is to design a mesh network where every node is equipped with a general OXC without wavelength converters, that is, the wavelength continuity constraint must be satisfied. The main features of the optimisation tasks presented here are that duct costs are included and wavelength assignment is integrated with path optimisation and duct placement. For simplicity we consider only link costs in the optimisation tasks of this chapter.

After defining the network components and optimisation tasks in the next section, we describe an algorithm for obtaining a set of "promising" paths that can help reduce the greenfield network cost by minimising both fibre and duct usage. These paths are subsequently used for network optimisation by ILP and two instances of metaheuristics: simulated annealing (SAN) and simulated allocation (SAL). The ILP programs exist in two flavours: arc-flow and link-path formulations, where the former produces optimal results and the latter produces results based on the quality of the set of promising paths. The reason for introducing heuristics is to be able to exceed the very small network sizes ILP solvers are able to handle within reasonable time: The heuristics do not guarantee optimal results but consider iteratively a large amount of solutions, attempting to find a solution with a value as close to the optimal as possible. The promising path generator

(PPG) and the SAL+SAN heuristics are collectively referred to as the *explain tool.*

We evaluate empirically the promising path generator (PPG) algorithm by comparing its results with optimal results obtained from ILP programs, using some small example networks. The heuristics are evaluated empirically by running them on three example networks, two of which are somewhat larger.

Finally, we show a few graphs which demonstrate that the heuristics stabilise at their final solutions within a reasonably short time, and we draw some conclusions from this chapter in Section 2.8.

2.1 Greenfield WDM network planning

Several network design scenarios exist, characterised by what network parameters are known (e.g., node positions) and what parameters can be extended (e.g., link equipment: fibres). If the node positions are unknown, it is a *totally greenfield* network design scenario in which the network operator must decide where to place the nodes based on knowledge of where the customers are located. We will however always assume that the node locations are known, and that we are given some new traffic demands that must be satisfied.

Network design scenarios can be classified as shown in Table 2.1. The most

Nodes		*Links*				*Traffic*	
equipment deployed	new eqp. allowed	ducts exist	new ducts allowed	fibres deployed	new fibres allowed	existing routed	*Network design problem type*
−	+	−	+	−	+	−	(Node restricted) **greenfield**
−	+	+	−	−	+	−	**Duct restricted greenfield**
+	−	+	−	+	−	−	Totally capacitated routing
+	−	+	−	+	−	+	Totally capacitated routing-extension
+	−	+	−	+	+	−	Node cap. routing & link deployment-ext.
+	−	+	−	+	+	+	Node cap. routing-ext. & link deplmt-ext.
+	+	+	−	+	−	−	(Link) capacitated routing
+	+	+	−	+	−	+	Link capacitated routing-extension
+	+	+	−	+	+	−	Duct restricted routing & deployment-ext.
+	+	+	−	+	+	+	Duct restricted routing-ext. & deplmt-ext.
+	+	+	+	+	+	−	Routing & deployment-extension
+	+	+	+	+	+	+	(General) **extension**

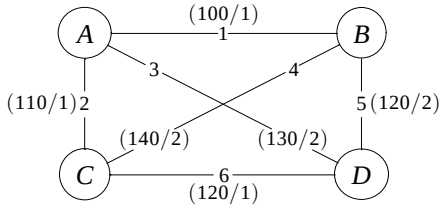
Table 2.1: Classification of various network design scenarios

common scenarios are the greenfield, the duct restricted greenfield and the extension network design scenarios. In the remaining part of this chapter we will consider the greenfield scenario, that is, the nodes and some traffic is given, and the goal is to design the placement and capacities of the links.

Nodes. Implicitly we assume that the OXCs in the N nodes have unbounded switching capacity, and we assume explicitly that no nodes are capable of wavelength conversion so the wavelength continuity constraint must be satisfied.

Links. Each link consists of one duct containing a number of fibre pairs, each of which is able to carry W wavelengths in each direction.

We will speak of the *potential links*, which is a set of L as of yet non-existing links from which the actual network links must be chosen during the design process. An example network with given nodes and potential links is shown in Figure 2.1. In the cost model used here, the cost of link l is $C_l^{\text{link}} = C_l^{\text{duct}} + f_l \cdot C_l^{\text{fibre}}$, where



	A	B	C	D
A		7	0	0
B	0		0	5
C	0	0		4
D	0	0	0	

Figure 2.1: WDM network example with 6 links and a traffic demand matrix containing 3 demands. Each links is numbered and labelled with (duct/fibre) cost

f_l is the number of fibres deployed on link l, and the cost of the whole network is just the sum of all the link costs. This also implies that node costs can be modeled if they are linear in the number of connected fibres, that is, only depend on the number of fibre ports. In this case the cost of the two fibre ports at the ends of a link can simply be added to the fibre cost before optimising; thus we assume that any fibre port cost is included in C_l^{fibre}.

Traffic. The traffic is static and given as a set of D traffic demands. Demand number d consists of a source node n_d^{src} and a destination node n_d^{dst} but as described previously the traffic is symmetric. Further, each demand d has a traffic volume V_d, measured in number of wavelengths. For simplicity we do not set any hop limit or length limit on the path(s) which supplies a demand.

Protection. We consider various different ways of protecting the traffic from network failures. In general we can consider the network links to be in state $s \in \{0, 1, \ldots, S\}$. We let state $s = 0$ denote the nominal state, i.e. a fully functioning network, and set $S = L$, such that state $s > 0$ is a failure state in which link s is

broken. This way we consider only *single link failures*, but it is straightforward to extend this framework to multiple failures or node failures.

Given a set of paths all supplying the same demand, that is, with identical source and destination nodes, we say that they are *(single link/node) failure resistant* if it is such that no matter which single link / node failure occurs, there are still some working paths in the set. Given a set of failure states, if for every pair of paths in the path set and every failure state at least one path is unaffected by the failure, we say that the set of paths is *(link/node) failure disjoint*. The set of paths shown in Figures 1.6(a)–(c) on page 17 is failure resistant, and the set of paths in Figures 1.6(b) and (c) is failure disjoint; note that the shortest path is not necessarily in the failure disjoint path set. If there exists a failure resistant set of paths supplying a demand, there also exists a failure disjoint set of paths supplying this demand. The paths used when there are no failures in the network are called *primary* paths, and additional paths used in case of failures are called *backup* paths.

In the present work we exclusively consider $m : n$ path protection, protecting the traffic against failures in the following ways:

Nominal design (ND): This is in fact the unprotected case, in which no spare backup capacity is deployed in the network.

Total rerouting protection (TRP): In each failure state the entire network is rerouted to satisfy the demands without using the broken links / nodes.

Path diversity protection (PDP): Each traffic demand is supplied by several failure resistant paths in such a way that no matter which failure occurs, the remaining working paths have sufficient total capacity to fulfil the demand.

Single backup path protection (SBP): Each lightpath that supplies a traffic demand is assigned a specific, failure disjoint backup lightpath which is used in all failure states that affect the primary lightpath.

Single backup path protection with fixed wavelength (SBP+FW): As SBP, but the backup lightpath must use the same wavelength as the primary lightpath.

Single backup path protection with non-reusable primary capacity (SBP+NRPC): As SBP, but the primary capacity unaffected by the failure (e.g., the wavelength on the top middle link in Figures 1.6(b) and (c)) cannot be reused for any other backup paths.

Single backup path protection with fixed wavelength and non-reusable primary capacity (SBP+FW+NRPC): A combination of the two preceding design scenarios.

TRP is the protection method requiring least spare capacity, so the cost found by TRP is a lower bound on the costs found in the other protection scenarios. However, TRP is normally not used in real-world networks because of the major signalling and switching complexity needed to reroute all demands, including those whose primary paths are not failed.

2.2 Shortest, disjoint and backup paths

Given a network as a set of nodes and a set of links, each link annotated with a cost, we can consider the following tasks for a given traffic demand between a source and destination node:

k_s shortest paths: Find a set of k_s paths from the source to the destination node with least cost. If fewer than k_s such paths exist, find as many as possible.

k_d disjoint paths: Find a set of k_d link disjoint paths such that the total cost of the used links is minimal. If fewer than k_d such paths exist, find as many as possible.

k_b backup paths: Removing temporarily from the network the links constituting the k_s shortest paths, find a set of k_b shortest paths from the source to the destination node with least cost. If fewer than k_b such paths exist, find as many as possible.

The shortest paths are useful for routing demands in the ND scenario where no failure-disjoint backup paths are needed, and the disjoint paths are useful for the protection scenarios. The backup paths typically "take the long way round" from the source to destination node, and are useful when trying to design a network with ring structures, because each primary route typically uses one of the shortest paths, and the secondary route one of the backup paths.

Note that the disjoint paths cannot in general be found simply by iteratively finding and removing the shortest path. An example where the shortest path is not one of the two disjoint paths is shown in Figure 2.2 on the facing page.

Disjoint paths, constituted by a set of edges P, can be found in a directed network by a simple algorithm:

1. Let $i \leftarrow 1$ and $P \leftarrow \{\}$
2. Find the shortest path p; if it does not exist (there is no path from source to destination), stop.
3. Let $P \leftarrow P \uplus p$, that is, for each edge in p, if it already is in P, remove it from P, otherwise add it to P.

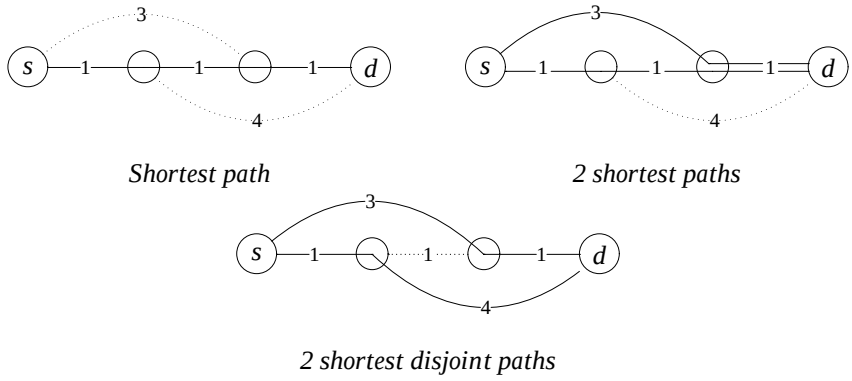


Figure 2.2: The shortest path is not one of the 2 shortest *disjoint* paths

4. Reverse the edges of the shortest path and negate the sign of the costs on the path links.

5. Let $i \leftarrow i+1$; if $i < k_d$ goto Step 2, else stop.

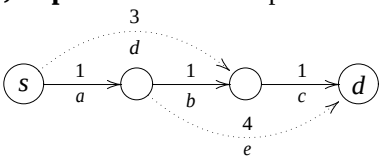
Note that the algorithm requires a shortest path algorithm which can handle links with negative costs—e.g., the Bellman-Ford algorithm. An example of how the disjoint paths algorithm works is shown in Figure 2.3 on the next page.

Algorithms for finding shortest paths are given in the literature (Yen, 1971; Suurballe, 1974; de Azevedo et al., 1994). Finding shortest paths in networks with negative-cost links by using the Bellman-Ford algorithm is of complexity $O(N \cdot L)$, so the algorithm for finding k_d disjoint paths can be implemented to run in time $O(k_d(N \cdot L + P))$, where P is the upper bound on the path lengths. The algorithm for finding k_s shortest paths can be implemented to run in time $O(k_s N^3)$ (Yen, 1971), and similarly finding k_b backup paths runs in time $O(k_b N^3)$. In total, assuming no more than a constant number of links between each node pair, finding the k_s, k_b, k_d paths takes time $O((k_s + k_d + k_b)N^3)$.

2.3 Promising path generation

Although the greenfield scenario provides complete freedom for the network designer to optimise the network by locating links wherever it seems best, it adds the complexity of link selection to the optimisation task. In many cases this added complexity makes the optimisation process prohibitively slow for realistic networks. Instead of optimising *both* the placement of links *and* the routing of traffic—

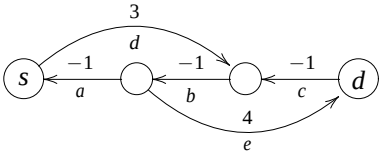
$i = 1$**, step 2:** Find shortest path:



$p = \{a, b, c\}$ $P = \{\}$

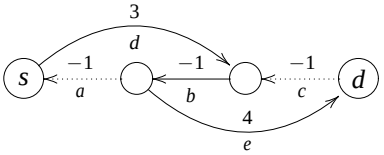
$i = 1$**, step 3:** Exor path with P: $P \leftarrow \{\} \uplus \{a, b, c\} = \{a, b, c\}$

$i = 1$**, step 4:** Reverse all shortest paths links, negating costs:



$P = \{a, b, c\}$

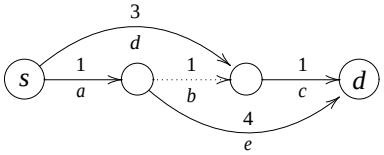
$i = 2$**, step 2:** Find shortest path:



$p = \{d, b, e\}$ $P = \{a, b, c\}$

$i = 2$**, step 3:** Exor path with P: $P \leftarrow \{a, b, c\} \uplus \{d, b, e\} = \{a, c, d, e\}$

Result:



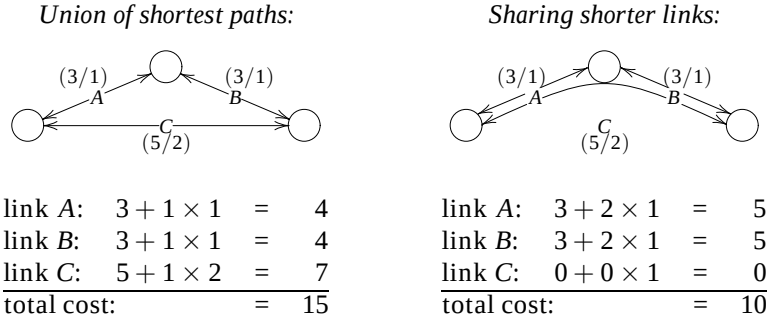
$P = \{a, c, d, e\}$

Figure 2.3: Example of finding the 2 disjoint shortest paths

an integrated optimisation approach—the problem can be split into two stages, each of manageable complexity. In the first stage a set of promising links is extracted from the set of all potential links, based on rough estimates of their cost, and their usefulness for the traffic demands. At the same time some promising paths—i.e., cost efficient paths—for each demand are computed. In the second stage, the network is optimised by only choosing paths from the set of promising paths.

We call the first stage *promising path generation* (PPG). Given the set of traffic demands and the cost of constructing each link in the network, we must select a subset of these links. Lee et al. (1998a) present an algorithm for this link and path selection, based on reducing the variance of the link loads, which is slightly similar to the algorithm we present in the following.

As a first attempt, one might compute the shortest path for each demand and then select the union of the used links. This strategy is no good, however, if there is an initial cost for constructing the duct of each link: Too many long links are selected, cf. Figure 2.4, where we assume traffic demands corresponding to 1 fibre between all node pairs. We can see that the best choice of paths and links depends



link A:	$3 + 1 \times 1$	=	4
link B:	$3 + 1 \times 1$	=	4
link C:	$5 + 1 \times 2$	=	7
total cost:		=	15

link A:	$3 + 2 \times 1$	=	5
link B:	$3 + 2 \times 1$	=	5
link C:	$0 + 0 \times 1$	=	0
total cost:		=	10

Figure 2.4: The union of the shortest paths include too many direct, long links. Duct / fibre costs are shown next to the links.

not only on the cost of the ducts and fibres, but also on the traffic to be transported: If there is a lot of traffic between two distant nodes, it may pay to construct a direct link, rather than hopping via several intermediate links. As the direct link is usually slightly shorter than the sum of the length of the intermediate links, the fibre cost will be smaller. Thus we must devise an algorithm for selecting links that strikes the right balance between the two extremes: using a lot of direct links (producing a fully connected mesh) and sending all long-distance traffic along paths with many hops (producing more or less a minimum spanning tree).

We find this balance by the following algorithm: We initially sort the traffic demands ascendingly according to the cost of constructing a direct link (i.e., the pure duct cost). To each link we assign a *traffic load* and *cost per traffic unit*, and then use an algorithm to iteratively update these values by routing the traffic demands one by one. Whenever a traffic demand is routed along a path, the traffic load on each link is incremented accordingly, and its cost per traffic unit is recalculated, based on the new traffic load. Finally, we can use the cost per traffic values as "link costs" in a path finding algorithm. The whole algorithm is sketched in Figure 2.5. The function $\mathrm{route}(d, \mathit{costpertraffic}, k_s, k_d, k_b)$ is used to find paths for demand d,

```
PromisingPaths(links, cost^duct, cost^fibre, demands, k_s, k_d, k_b) =
  for d ∈ demands do                                  /* Initialise cost^direct   */
    L ← {l ∈ links | src_l = src_d ∧ dst_l = dst_d}    /* Find direct links        */
    if L ≠ {} then
      cost_d^direct ← min{cost_l^duct | l ∈ L}
    else
      cost_d^direct ← ∞
  sort demands according to cost^direct
  for l ∈ links do traffic_l ← 0                      /* Initialise traffic       */
  for l ∈ links do costpertraffic_l ← cost_l^duct     /* Initialise costpertraffic*/
  for d ∈ demands do                                  /* Main loop:               */
    paths ← route(d, costpertraffic, 1, 2, 1)         /* Route traffic demand     */
    for p ∈ paths do
      for l ∈ p do
        traffic_l ← traffic_l + volume_d              /* Update traffic           */
        costpertraffic_l ← cost(l, traffic_l)/traffic_l /* Update costpertraffic  */
  for d ∈ demands do
    R_d ← route(d, costpertraffic, k_s, k_d, k_b)     /* Perform final routing    */
  return {R_d | d ∈ demands}
```

Figure 2.5: Promising path generation algorithm

based on the link cost values in *costpertraffic*; up to k_s shortest, k_d disjoint and k_b backup paths are found as described in Section 2.2. The cost of carrying $\mathit{traffic}_l$ wavelengths on link l is calculated by the function $\mathrm{cost}(l, \mathit{traffic}_l)$, based on the duct and fibre costs, and the number of wavelengths per fibre, W.

One of the effects that makes the algorithm work is that short links between neighbouring nodes tend to be loaded with traffic before demands between more distant nodes are routed, thus enabling them to use some of the short links instead of creating long, direct links.

Looking at the algorithm, we then see that it is important that there are traffic demands between "neighbouring nodes" so that short links between these nodes are created. We ensure this by creating a dummy demand with volume zero for any node pair that does not have an explicit traffic demand. A further slight improvement of the algorithm can be made by temporarily adjusting the *costpertraffic* values as if $volume_d$ wavelengths were added to every network link just before calling the route function.

Regarding the complexity of the PPG algorithm, we see that it is dominated by the sorting in time $O(D \log D)$, traversing the set of links in time $O(L)$, updating the *traffic* matrix in time $O(D \cdot K \cdot P)$ and the routing of paths in time $O(D \cdot KN^3)$, where $K = k_s + k_d + k_b$ and P is the upper bound on the path length. In total, we find the algorithm runs in time $O(D \log D + L + D \cdot K \cdot P + D \cdot KN^3) = O(DKN^3)$.

2.4 Optimising by integer linear programming

The network problems we have been considering are variants of the multicommodity flow problem that has been intensely studied in the literature (Srinivasan, 1997; Lee et al., 1998a; Ahuja et al., 1993). In this section we cast the network design problems in mathematical formulations as ILP programs.

ILP programs can act as rigorous definitions of the problems, but are especially useful as benchmarks for optimised network design heuristics, because they can be passed directly to ILP optimisers. The optimisers we use here are GAMS and CPLEX: GAMS is a preprocessor which takes the formulations we give, re-casts them in a shape that CPLEX can handle and performs some simple preoptimisations, before passing them to CPLEX which is the real optimisation tool.

There are two major ways of expressing a multicommodity flow problem in ILP programs: as an *arc-flow* or *link-path* formulation. The major difference is that in the arc-flow formulation it is up to the optimiser to construct the paths that must supply the demands, whereas in the link-path formulation we must precompute some paths for each demand, and the optimiser is only able to select between these precomputed paths. It then follows that the optimal values found by arc-flow based optimisation are always at least as good as the optimal values found by link-path based optimisation—only if *all possible* paths are given as input to the link-path formulation can we be sure of obtaining the same results. Typically, fewer paths are sufficient for getting a good link-path solution, and this is often preferred, as running the optimiser on the arc-flow formulation usually consumes much more computer memory and CPU time.

2.4.1 Arc-flow formulation

In the arc-flow formulation we consider the network as a set of N nodes: at each node some wavelengths enter it, either from neighbouring nodes or from a traffic source located in the node. The exact same amount must leave the node, either to neighbouring nodes or to a traffic destination located in the node: this is the *flow conservation principle.* These flows are then indexed by the demand they supply, and in our case also the wavelength they use. The traffic flows along directed edges, and the directed edge from node n to node m is carried together with the edge from node m to node n in a link named $\{n, m\}$. This implies that it is not possible to have more than one link between two nodes in the arc-flow formulation.

The optimisation process for the arc-flow formulation is shown schematically in Figure 2.6. A simple generator produces the ILP program, based on the traffic

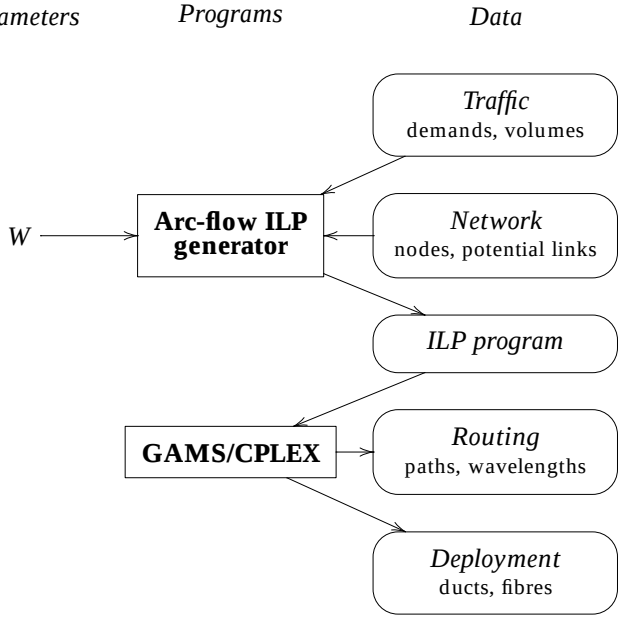


Figure 2.6: Optimising WDM networks using the arc-flow based ILP method

and network data, and then the ILP program is passed to the GAMS/CPLEX optimiser, which produces the final optimisation result.

In the following we present arc-flow formulations for the ND, TRP, PDP and

SBP problems. They are inspired by the formulation given by Doshi et al. (1999), who introduce a set of *states* $0, \ldots, S$, and for each state a set $\mathcal{E}_s$ of *failed edges*. Unless otherwise stated, each state s corresponds to exactly one edge failure, but the state formulation is a general approach that also allows modelling node failures by letting all the edges incident with a node failing in state s be in $\mathcal{E}_s$. Conventionally, $s = 0$ is considered to be the nominal state where there are no failures, that is, $\mathcal{E}_0 = \{\}$.

All the various network design problems share the following indexes, constants, basic variables and objective function:
Indexes:

$d \in \{1, \ldots, D\}$	Traffic demands
$s \in \{0, \ldots, S\}$	Network states
$c \in \{1, \ldots, W\}$	Wavelengths (colours)
$m, n \in \{1, \ldots, N\}$	Network nodes
$(nm) \in \mathcal{E} \subseteq \{1, \ldots, N\}^2$	Directed edges
$(nm) \in \mathcal{E}_s \subset \mathcal{E}$	Directed edges affected in state s
$\{n, m\} \in \mathcal{L} \subseteq \mathcal{P}(\{1, \ldots, N\})$	Undirected links

Constants:

$V_d^s \in \mathbb{N}_0$	Volume of lightpaths to be realised for demand d in state s
$n_d^{\text{src}} \in \{1, \ldots, N\}$	Index of the source node for demand d
$n_d^{\text{dst}} \in \{1, \ldots, N\}$	Index of the destination node for demand d
$C_{\{n,m\}}^{\text{duct}} \in \mathbb{R}_+$	Cost of constructing the duct on link $\{n, m\}$
$C_{\{n,m\}}^{\text{fibre}} \in \mathbb{R}_+$	Cost of deploying a fibre pair on link $\{n, m\}$

Basic variables:

$u_{\{n,m\}} \in \mathbb{R}_+$	Number of required fibre pairs on link $\{n, m\}$
$\delta_{\{n,m\}} \in \{0, 1\}$	Number of required ducts on link $\{n, m\}$

Objective:
Minimise $cost_{total} = \sum_{\{n,m\}} C_{\{n,m\}}^{\text{fibre}} \cdot u_{\{n,m\}} + C_{\{n,m\}}^{\text{duct}} \cdot \delta_{\{n,m\}}$.

Notation: Note that $\mathbb{R}_+$ denotes the set of nonnegative reals, and that variable $u_{\{n,m\}}$ is declared as a nonnegative real because this can speed up ILP optimisers. In the following ILP programs, such nonnegative real variables are constrained

by integer variables, and will thus always represent integer values in optimal solutions. Furthermore, the quantification "$m : (nm) \in \mathcal{E}$" denotes "all m for which there exists an edge (nm) in $\mathcal{E}$."

Nominal network design problem (ND)

In this design problem we just have to supply a volume of V_d^0 wavelengths for each demand d.

Additional variables:

$x_{dnm}^c \in \mathbb{N}_0$	Flow of demand d on wavelength c along edge (nm)
$v_d^c \in \mathbb{R}_+$	Volume of demand d carried on wavelength c

Constraints:

$\sum_c v_d^c = V_d^0,$	$\forall d$	The total volume of demand d must be supplied by lightpaths of various wavelengths
$\sum_{m:(nm)\in\mathcal{E}} x_{dnm}^c - \sum_{m:(mn)\in\mathcal{E}} x_{dmn}^c = \begin{cases} v_d^c, & \text{if } n = n_d^{\text{src}} \\ -v_d^c, & \text{if } n = n_d^{\text{dst}} \\ 0, & \text{otherwise} \end{cases}$	$\forall c, d, n$	Flow conservation for paths: what goes into node n on wavelength c must come out again
$\sum_d x_{dnm}^c + x_{dmn}^c \le u_{\{n,m\}}$	$\forall c, \{n, m\}$	Calculate the required number of fibres on any wavelength c
$\delta_{\{n,m\}} = 0 \Rightarrow u_{\{n,m\}} = 0$	$\forall \{n, m\}$	Without a duct, there can be no fibres on link $\{n, m\}$

Total rerouting protection network design problem (TRP)

The TRP design problem is constructed by extending the nominal design problem to include the network failure states. Naturally, if the source or destination node of demand d fails in state s we cannot in any way supply the demand, in which case we set $V_d^s = 0$. There may also be other reasons for a network operator to let $V_d^s \neq V_d^0$.

Additional variables:

$x_{dnm}^{cs} \in \mathbb{N}_0$	Flow of demand d on wavelength c along edge (nm) in state s
$v_d^{cs} \in \mathbb{R}_+$	Volume of demand d carried on primary wavelength c in state s

Constraints:

$\sum_c v_d^{cs} \geq V_d^s,$	$\forall d, s$	The total volume of demand d in each state must be supplied by lightpaths of various wavelengths
$\sum_{m:(nm)\in\mathcal{E}\setminus\mathcal{E}_s} x_{dnm}^{cs} - \sum_{m:(mn)\in\mathcal{E}\setminus\mathcal{E}_s} x_{dmn}^{cs} = \begin{cases} v_d^{cs}, \text{ if } n = n_d^{\text{src}} \\ -v_d^{cs}, \text{ if } n = n_d^{\text{dst}} \\ 0, \text{ otherwise} \end{cases}$	$\forall c, d, s, n$	Flow conservation for paths: what goes into node n on wavelength c in state s must come out again
$\sum_d x_{dnm}^{cs} + x_{dmn}^{cs} \leq u_{\{n,m\}}$	$\forall c, s, \{n, m\}$	Calculate the required number of fibres in any state s on any wavelength c
$\delta_{\{n,m\}} = 0 \Rightarrow u_{\{n,m\}} = 0$	$\forall \{n, m\}$	Without a duct, there can be no fibres on link $\{n, m\}$

Path diversity protection network design problem (PDP)

In the PDP design problem we use x_{dnm}^{cs} to keep track of the flows in every state s, and find the required capacity t_{dnm}^{c} as the maximum of x_{dnm}^{cs} over all states.

Additional variables:

$x_{dnm}^{cs} \in \mathbb{N}_0$	Flow of demand d on wavelength c in state s along edge (nm)
$t_{dnm}^{c} \in \mathbb{R}_+$	Required capacity for demand d on wavelength c along edge (nm)
$v_d^{cs} \in \mathbb{R}_+$	Volume of demand d carried on primary wavelength c in state s

Constraints:

$\sum_c v_d^{cs} \geq V_d^s,$	$\forall d, s$	The total volume of demand d in each state must be supplied by lightpaths of various wavelengths
$\sum_{m:(nm)\in\mathcal{E}\setminus\mathcal{E}_s} x_{dnm}^{cs} - \sum_{m:(mn)\in\mathcal{E}\setminus\mathcal{E}_s} x_{dmn}^{cs} = \begin{cases} v_d^{cs}, \text{ if } n = n_d^{\text{src}} \\ -v_d^{cs}, \text{ if } n = n_d^{\text{dst}} \\ 0, \text{ otherwise} \end{cases}$	$\forall c, d, s, n$	Flow conservation for paths: what goes into node n on wavelength c in state s must come out again
$x_{dnm}^{cs} \leq t_{dnm}^{c}$	$\forall c, d, s, (nm)$	Calculate the required capacity for demand d on wavelength c along edge (nm)
$\sum_d t_{dnm}^{c} + t_{dmn}^{c} \leq u_{\{n,m\}}$	$\forall c, \{n, m\}$	Calculate the required number of fibres on any wavelength c
$\delta_{\{n,m\}} = 0 \Rightarrow u_{\{n,m\}} = 0$	$\forall \{n, m\}$	Without a duct, there can be no fibres on link $\{n, m\}$

Single backup path network design problem (SBP)

The SBP problem requires mapping a backup lightpath to each primary lightpath, so in the arc-flow formulation we must explicitly represent the flow of each single lightpath along each link, given by the binary variable x_{dnm}^{c}. For this reason, if we want the optimal routing, a traffic demand of volume V_d^0 must be split into V_d^0 individual demands, each of just one wavelength. However, if the volume for each demand is large, we might want to *bundle* the lightpaths to keep the ILP program small for optimisation efficiency reasons. In that case all V_d^0 lightpaths will use the same primary path, and will all be switched to the same backup path.

Additional variables:

$x_{dnm}^{c} \in \{0, 1\}$	Primary flow of demand d on wavelength c, edge (nm)
$y_{dnm}^{c} \in \{0, 1\}$	Backup flow of demand d on wavelength c, edge (nm)

$t^{cs}_{dnm} \in \mathbb{R}_+$	Primary capacity required for d on wavelength c, edge (nm) in state s
$z^{cs}_{dnm} \in \mathbb{R}_+$	Backup capacity required for d on wavelength c, edge (nm) in state s
$v^c_d \in \mathbb{R}_+$	Volume of demand d carried on primary wavelength c
$w^c_d \in \mathbb{R}_+$	Volume of demand d carried on backup wavelength c
$\varphi^s_d \in \{0,1\}$	Demand d is affected by state s

Constraints:

$\sum_c v^c_d = 1$	$\forall d$	Exactly one wavelength is used for supplying each demand
$\sum_c w^c_d = 1$	$\forall d$	Exactly one wavelength is used for backing up each demand
$\sum_{m:(nm)\in\mathcal{E}} x^c_{dnm} - \sum_{m:(mn)\in\mathcal{E}} x^c_{dmn} = \begin{cases} v^c_d, & \text{if } n = n^{\text{src}}_d \\ -v^c_d, & \text{if } n = n^{\text{dst}}_d \\ 0, & \text{otherwise} \end{cases}$	$\forall c, d, n$	Flow conservation for primary paths: what goes into node n on wavelength c must come out again
$\sum_{m:(nm)\in\mathcal{E}} y^c_{dnm} - \sum_{m:(mn)\in\mathcal{E}} y^c_{dmn} = \begin{cases} w^c_d, & \text{if } n = n^{\text{src}}_d \\ -w^c_d, & \text{if } n = n^{\text{dst}}_d \\ 0, & \text{otherwise} \end{cases}$	$\forall c, d, n$	Flow conservation for backup paths: what goes into node n on wavelength c must come out again
$\varphi^s_d = 0 \Leftrightarrow \sum_c \sum_{(nm)\in\mathcal{E}_s} x^c_{dnm} = 0$	$\forall d, s$	Demand d is unaffected in state s iff no primary edges are in failure set $\mathcal{E}_s$
$V^s_d \varphi^s_d \geq 1 \Rightarrow \sum_c \sum_{(nm)\in\mathcal{E}_s} y^c_{dnm} = 0$	$\forall d, s$	If demand d should be restored in state s, no backup edges may be in failure set $\mathcal{E}_s$
$\varphi^s_d = 0 \Rightarrow t^{cs}_{dnm} = V^s_d x^c_{dnm}$	$\forall c, d, s, (nm)$	If demand d is not switched to a backup edge (nm) in state s, the primary flow t^{cs}_{dnm} is $V^s_d x^c_{dnm}$

$\varphi_d^s \geq 1 \Rightarrow z_{dnm}^{cs} = V_d^s y_{dnm}^c$	$\forall c, d, s, (nm)$	If demand d is switched to a backup edge (nm) in state s, the backup flow z_{dnm}^{cs} is $V_d^s y_{dnm}^c$
$\sum_d t_{dnm}^{cs} + t_{dmn}^{cs}$ $+\sum_d z_{dnm}^{cs} + z_{dmn}^{cs} \leq u_{\{n,m\}}$	$\forall c, s, \{n, m\}$	Calculate the required number of fibres in any state s on any wavelength c
$\delta_{\{n,m\}} = 0 \Rightarrow u_{\{n,m\}} = 0$	$\forall \{n, m\}$	Without a duct, there can be no fibres on edge $\{n, m\}$

Single backup path design problem variations

To add the "fixed wavelength" condition to the ILP for the SBP problem, we simply replace all occurrences of w_d^c with v_d^c, and to add the "non-reusable primary capacity" condition we replace constraint

$$\varphi_d^s = 0 \Rightarrow t_{dnm}^{cs} = V_d^s x_{dnm}^c \qquad \text{with} \qquad t_{dnm}^{cs} = V_d^s x_{dnm}^c$$

Handling multiple link failures

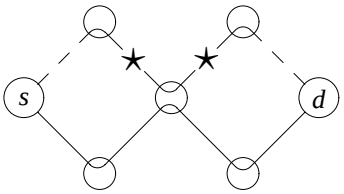
The formulations we have given in this section are also able to handle multiple link failures if it is at all possible, that is, for any given set of failed links $\mathcal{E}_s$, *if* the network connectivity is strong enough, the ILP programs presented here will be able to find a solution. For instance, consider the network shown in Figure 2.7 on the next page, with just one demand from s to d, and three failure states, one where two upper links fail, and one where two lower links fail, and one where an upper and lower link fail. Both TRP and PDP are able to handle all situations, in each case using the backup path shown. There exists no solution to the SBP problem in this case: no single combination of (primary,backup) path is failure disjoint. However, if we consider just failures states (a) and (b), the ILP formulation of SBP presented here is able to provide the correct solution. This is not the case for the ILP given by Doshi et al. (1999, p. 68):

$$\sum_{m:(nm)\in\mathcal{E}} x_{dnm} - \sum_{m:(mn)\in\mathcal{E}} x_{dmn} = \begin{cases} 1, & \text{if } n = n_d^{\text{src}} \\ -1, & \text{if } n = n_d^{\text{dst}} \\ 0, & \text{otherwise} \end{cases} \quad \forall d, n$$

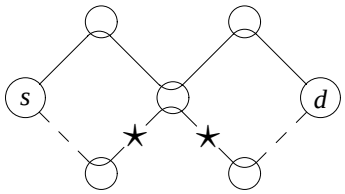
Flow conservation for primary paths: what goes into node n must come out again

$$\sum_{m:(nm)\in\mathcal{E}} y_{dnm} - \sum_{m:(mn)\in\mathcal{E}} y_{dmn} = \begin{cases} 1, & \text{if } n = n_d^{\text{src}} \\ -1, & \text{if } n = n_d^{\text{dst}} \\ 0, & \text{otherwise} \end{cases} \quad \forall d, n$$

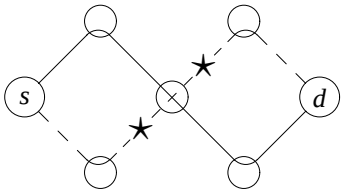
Flow conservation for backup paths: what goes into node n must come out again



(a) Two upper links fail



(b) Two lower links fail



(c) An upper and lower link fails

Figure 2.7: Multiple link failures

$$\begin{array}{r} \sum_{m:(nm)\in\mathcal{E}\setminus\mathcal{E}_s} z^s_{dnm} \\ -\sum_{m:(mn)\in\mathcal{E}\setminus\mathcal{E}_s} z^s_{dmn} \end{array} = \begin{cases} \sum_{(kl)\in\mathcal{E}_s} x_{dkl}, & \text{if } n = n_d^{\text{src}} \\ -\sum_{(kl)\in\mathcal{E}_s} x_{dkl}, & \text{if } n = n_d^{\text{dst}} \\ 0, & \text{otherwise} \end{cases} \quad \forall d, n, s$$

This is supposed to ensure that the backup path is used exactly when the primary path fails

$$z^s_{dnm} \leq y_{dnm} \qquad \forall d, s, (nm) \notin \mathcal{E}_s$$

Ensure that the z variables use the same backup path in all failure states

$$\sum_d x_{dnm} + \sum_d z^s_{dnm} \leq u_{(nm)} \qquad \forall s, (nm) \notin \mathcal{E}_s$$

Calculate the required number of fibres in any state s

If the lower path is chosen as the primary path, we find that $\sum_{(kl)\in\mathcal{E}_s} x_{dkl} = 2$, which prevents a feasible solution to the third set of equations in failure state (b). Further, their ILP as stated is not able to handle source or destination node failures.

2.4.2 Link-path formulation

In the link-path formulation we consider the network as a set of L undirected links, and for each traffic demand d we precalculate a set of P_d potential primary paths. Note that as there is no direct concept of nodes involved, it is possible to model a network with more than one link between two nodes. For each primary path p, we calculate a set of Q_d^p potential backup paths where each path is failure disjoint with p. As in the arc-flow formulation, we consider a set of network states $s = 0, \ldots, S$, where each state conventionally corresponds to a single link failure, and state $s = 0$ to no failures.

The optimisation process for the link-path formulation is shown schematically in Figure 2.8. A preprocessor takes the traffic and the network and generates the

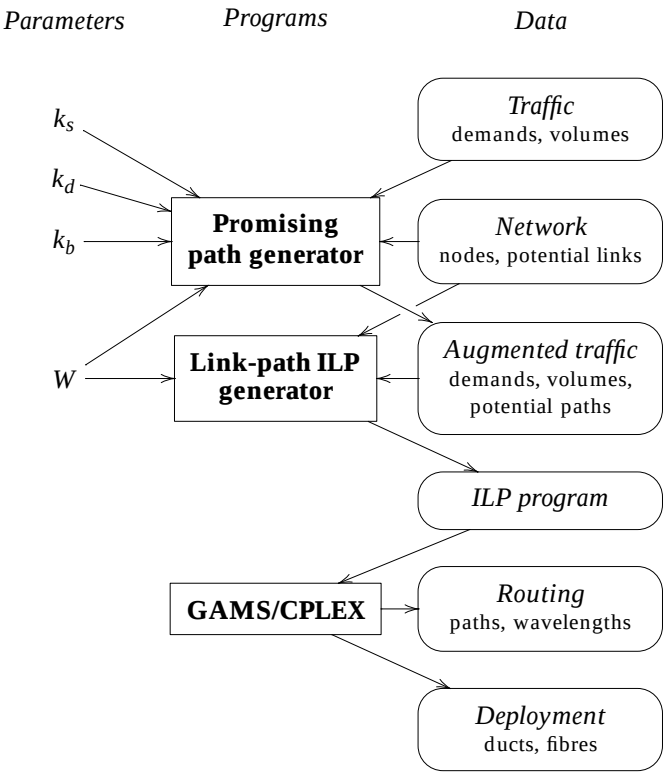


Figure 2.8: Optimising WDM networks using the link-path based ILP method

sets of primary and secondary paths. These are then passed to the ILP generator that produces the ILP program, which the GAMS / CPLEX optimiser takes as input to produce the final optimisation result.

Indexes:

$d \in \{1,\ldots,D\}$	Traffic demands
$s \in \{0,\ldots,S\}$	Network states
$c \in \{1,\ldots,W\}$	Wavelengths (colours)
$k \in \{1,\ldots,W\}$	Backup wavelengths
$l \in \{1,\ldots,L\}$	Network links
$p \in \{1,\ldots,P_d\}$	Primary paths realising demand d
$q \in \{1,\ldots,Q_d^p\}$	Backup paths protecting demand d, path p

Constants:

$V_d^s \in \mathbb{N}_0$	Volume of lightpaths to be realised for demand d in state s
$A_{ldp} \in \{0,1\}$	Primary link-path incidence coefficient: $A_{ldp} = 1$ if link l belongs to path p realizing demand d, $A_{ldp} = 0$ otherwise
$B_{ldpq} \in \{0,1\}$	Backup link-path incidence coefficient: $B_{ldpq} = 1$ if link l belongs to backup path q which can protect path p of demand d, $B_{ldpq} = 0$ otherwise
$F_l^s \in \{0,1\}$	Link failure coefficient: $F_l^s = 0$ if link l is failed in state s, $F_l^s = 1$ if it works
$G_{dp}^s \in \{0,1\}$	Path failure coefficient: $G_{dp}^s = 0$ if path p of demand d is failed in state s, $G_{dp}^s = 1$ otherwise, i.e., $G_{dp}^s = \prod_{l:A_{ldp}=1} F_l^s$
$C_l^{\text{duct}} \in \mathbb{R}_+$	Cost of constructing the duct on link l
$C_l^{\text{fibre}} \in \mathbb{R}_+$	Cost of deploying a fibre pair on link l

Basic variables:

$u_l \in \mathbb{R}_+$	Number of required fibre pairs on link l
$\delta_l \in \{0,1\}$	Number of required ducts on link l

Objective:

Minimise $cost_{total} = \sum_l C_l^{\text{fibre}} \cdot u_l + C_l^{\text{duct}} \cdot \delta_l$.

The constants for the primary flows of example in Figure 2.1 on page 26 are shown in Figure 2.9 on the following page.

Links

l	between	C_l^{duct}	C_l^{fibre}
1	$A-B$	100	1
2	$A-C$	110	1
3	$A-D$	130	2
4	$B-C$	140	2
5	$B-D$	120	2
6	$C-D$	120	1

Demands

d	between	V_d^s	P_d
1	$A-B$	7	3
2	$B-D$	5	3
3	$C-D$	4	2

Paths

d	p	A_{ldp}						Q_d^p
1	1	1	0	0	0	0	0	2
	2	0	0	1	0	1	0	2
	3	0	1	0	0	1	1	1
2	1	0	0	0	0	1	0	2
	2	1	0	1	0	0	0	1
	3	0	0	0	1	0	1	1
3	1	0	0	0	0	0	1	3
	2	0	1	1	0	0	0	2
	$l=$	1	2	3	4	5	6	

Link failures

s	F_l^s					
0	1	1	1	1	1	1
1	0	1	1	1	1	1
2	1	0	1	1	1	1
3	1	1	0	1	1	1
4	1	1	1	0	1	1
5	1	1	1	1	0	1
6	1	1	1	1	1	0
$l=$	1	2	3	4	5	6

Path failures

d	p	G_{dp}^s						
1	1	1	0	1	1	1	1	1
	2	1	1	1	0	1	0	1
	3	1	1	0	1	1	0	0
2	1	1	1	1	1	1	0	1
	2	1	0	1	0	1	1	1
	3	1	1	1	1	0	1	0
3	1	1	1	1	1	1	1	0
	2	1	1	0	0	1	1	1
	$s=$	0	1	2	3	4	5	6

Backup paths

d	p	q	B_{ldpq}					
1	1	1	0	0	1	0	1	0
		2	0	1	0	0	1	1
	2	1	1	0	0	0	0	0
		2	0	1	0	1	0	0
	3	1	1	0	0	0	0	0
2	1	1	1	0	1	0	0	0
		2	0	0	0	1	0	1
	2	1	0	0	0	0	1	0
	3	1	0	0	0	0	1	0
3	1	1	0	1	1	0	0	0
		2	0	0	0	1	1	0
		3	1	1	0	0	1	0
	2	1	1	1	0	0	1	0
		2	0	0	0	1	1	0
		3	0	0	0	0	0	1
		$l=$	1	2	3	4	5	6

Figure 2.9: WDM network example from Figure 2.1 represented by link-path ILP parameters

Nominal network design problem (ND)

Additional variables:

$x_{dp}^c \in \mathbb{N}_0$	Flow of demand d in wavelength c on path p
$v_l^c \in \mathbb{R}_+$	Number of times the wavelength c is used on link l

Constraints:

$\sum_p \sum_c x_{dp}^c = V_d^0$	$\forall d$	The total volume of demand d must be supplied by lightpaths of various wavelengths
$\sum_d \sum_p A_{ldp} x_{dp}^c = v_l^c$	$\forall c, l$	Calculate v_l^c
$v_l^c \leq u_l$	$\forall c, l$	Calculate the required number of fibres on any wavelength c
$\delta_l = 0 \Rightarrow u_l = 0$	$\forall l$	Without a duct, there can be no fibres on link l

Total rerouting protection network design problem (TRP)

This network design problem is easily found by extending that of ND to allow for failure states s.

Additional variables:

$x_{dp}^{cs} \in \mathbb{N}_0$	Flow of demand d in wavelength c on path p in state s
$v_l^{cs} \in \mathbb{R}_+$	Number of times the wavelength c is used on link l in state s

Constraints:

$\sum_p G_{dp}^s \sum_c x_{dp}^{cs} \geq V_d^s$	$\forall d, s$	The total volume of demand d in each state s must be supplied by working lightpaths of various wavelengths
$\sum_d \sum_p A_{ldp} x_{dp}^{cs} = v_l^{cs}$	$\forall c, l, s$	Calculate v_l^{cs}
$v_l^{cs} \leq u_l$	$\forall c, l, s$	Calculate the required number of fibres on any wavelength c
$\delta_l = 0 \Rightarrow u_l = 0$	$\forall l$	Without a duct, there can be no fibres on link l

Path diversity network design problem (PDP)

This network design problem is identical to TRP, except the flow variable x and the link usage variable v are not indexed by the state s.

Additional variables:

$x_{dp}^c \in \mathbb{N}_0$	Flow of demand d in wavelength c on path p
$v_l^c \in \mathbb{R}_+$	Number of times the wavelength c is used on link l

Constraints:

$\sum_p G_{dp}^s \sum_c x_{dp}^c \geq V_d^s$	$\forall d, s$	The total volume of demand d in each state s must be supplied by working lightpaths of various wavelengths
$\sum_d \sum_p A_{ldp} x_{dp}^c = v_l^c$	$\forall c, l$	Calculate v_l^c
$v_l^c \leq u_l$	$\forall c, l$	Calculate the required number of fibres on any wavelength c
$\delta_l = 0 \Rightarrow u_l = 0$	$\forall l$	Without a duct, there can be no fibres on link l

Single backup path network design problem (SBP)

For this design problem we introduce a binary variable β_{dpq}^{ck} for keeping track of which backup lightpaths (q,k) are used by which primary lightpaths (p,c), and a backup flow variable y_{dpq}^{ck} that keeps track of the corresponding backup volume. Note that contrary to the arc-flow formulation, $V_d^s > 1$ does *not* bundle the lightpaths of demand d.

Additional variables:

$x_{dp}^c \in \mathbb{N}_0$	Primary flow of demand d in wavelength c on path p
$y_{dpq}^{ck} \in \mathbb{N}_0$	Backup flow on path q in wavelength k protecting primary flow x_{dp}^c
$\beta_{dpq}^{ck} \in \{0,1\}$	Backup indicator: $\beta_{dpq}^{ck} = 1$ if wavelength c on path p supplying demand d is backed up by wavelength k on path q
$v_l^{cs} \in \mathbb{R}_+$	Number of times the wavelength c is used on link l in state s

Constraints:

$\sum_p \sum_c x^c_{dp} \geq V^s_d$	$\forall d, s$	The total volume of demand d in each state s must be supplied by working lightpaths of various wavelengths
$\sum_q \sum_k y^{ck}_{dpq} = x^c_{dp}$	$\forall d, c, p$	Satisfy backup demands
$\sum_q \sum_k \beta^{ck}_{dpq} = 1$	$\forall d, c, p$	Each lightpath (c, p) must have exactly one backup lightpath
$\beta^{ck}_{dpq} = 0 \Rightarrow y^{ck}_{dpq} = 0$	$\forall d, c, p, k, q$	Ensure that x^c_{dp} uses only one backup lightpath
$\sum_d \sum_p \begin{pmatrix} G^s_{dp} A_{ldp} x^c_{dp} + \\ (1 - G^s_{dp}) \sum_q \sum_k B_{ldpq} y^{kc}_{dpq} \end{pmatrix} = v^{cs}_l$	$\forall s, c, l$	Calculate v^{cs}_l
$v^{cs}_l \leq u_l$	$\forall s, c, l$	Calculate the required number of fibres in any state s on any wavelength c
$\delta_l = 0 \Rightarrow u_l = 0$	$\forall l$	Without a duct, there can be no fibres on link l

Single backup path design problem variations

To add the "fixed wavelength" condition to the ILP for the SBP problem, we simply remove all instances of k and $\sum_k$, so for example β^{ck}_{dpq} is replaced with β^c_{dpq} and y^{ck}_{dpq} with y^c_{dpq}. To add the "non-reusable primary capacity" condition, we replace constraint

$$\sum_d \sum_p \begin{pmatrix} G^s_{dp} A_{ldp} x^c_{dp} + \\ (1 - G^s_{dp}) \sum_q \sum_k B_{ldpq} y^{kc}_{dpq} \end{pmatrix} = v^{cs}_l, \ \forall s, c, l \qquad \text{with}$$

$$\sum_d \sum_p \begin{pmatrix} A_{ldp} x^c_{dp} + \\ (1 - G^s_{dp}) \sum_q \sum_k B_{ldpq} y^{kc}_{dpq} \end{pmatrix} = v^{cs}_l, \ \forall s, c, l$$

2.4.3 Complexity blowup due to symmetry in ILP programs

The ILP programs given in the preceding sections use an index c to keep track of traffic flow for each wavelength, but strictly speaking this causes the solutions to contain *more* information than we need: We only need to know how *many times* each wavelength is needed, not actually *which* wavelength it is. Stated another

way, whenever we are given a solution in the form of values for x_{dp}^{c}, we can find $W!$ other solutions with exactly the same cost and routing by permuting the c indices. In fact we would consider them all identical, for we do not care which wavelength a specific demand uses, only that it is different from that of other lightpaths with which it has links in common.

This *symmetry problem* may seem like a mere curiosity, but it can in fact have diverse effects on the running time of the ILP solvers if they are unable to detect the symmetry. To circumvent the symmetry problem it is possible in the link-path formulation to express the wavelength constraints in a different way which does not identify each wavelength by index c. Instead, paths that do not have any links in common are grouped together as *independent sets* (of paths), and link capacity is supplied by indicating how many times each group is needed. In this way the wavelength of each group is not recorded, and wavelengths can be assigned in a straightforward way in a post-optimisation phase.

This trick of grouping paths into separate layers with different wavelengths has been used by several authors (Cinkler et al., 2000; Sato, 1996) and is known as the *independent set formulation* (Ramaswami and Sivarajan, 1995). It avoids the symmetry problem, but unfortunately the number of different independent sets can become exponentially large, so although the problem size has been reduced, one still needs to employ special techniques like column generation (Lee et al., 2000) to handle problems of realistic sizes.

2.5 Optimisation by metaheuristics

The ILP programs given in the preceding sections serve well as formal definitions of the network design problems, and they can also be used directly as input to ILP optimisers. However, they are all some variant of the multicommodity flow problem, which is known to be NP-complete, so for practical purposes using general ILP optimisers is not a viable approach. We turn therefore to heuristics, and in this section we will present two instances of metaheuristics that perform optimisation by stochastic algorithms: *simulated annealing* and *simulated allocation*.

The overall structure of the optimisation by these heuristics is sketched in Figure 2.10 on the next page. As seen from the figure, they both handle the problem in a link-path formulation, which requires promising paths to be generated for the optimiser to operate on.

Both heuristics operate on the same kind of data structures, and both solve the same problems: the ND problem and the SBP+NRPC problem, where we protect the traffic 100% against all single-link failures, that is, $V_d^0 = V_d^1 = \cdots = V_d^S$. The demand volume V_d^0 can be any nonnegative integer, and the supplying lightpaths

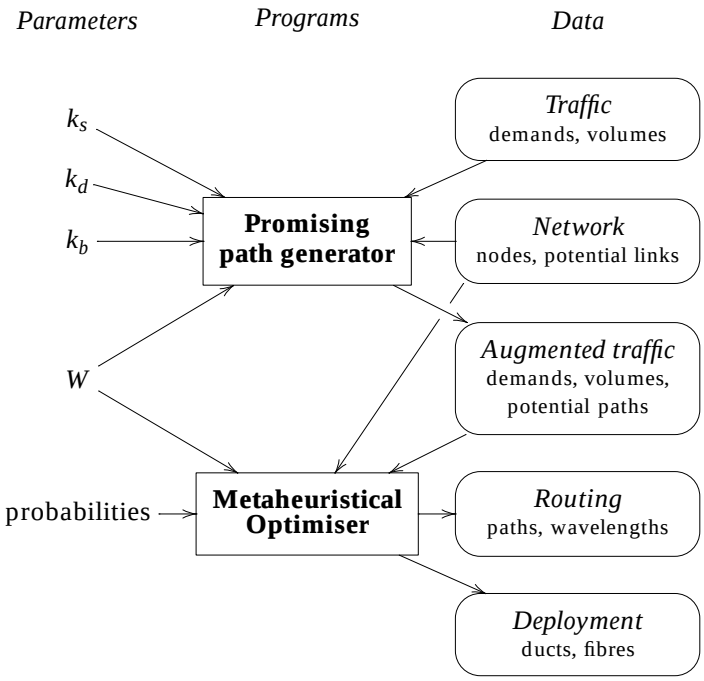


Figure 2.10: Optimising WDM networks using the simulated annealing or simulated allocation method

are *not* bundled.

2.5.1 Data structures

The main data structure used is a *solution*. A solution consists of a set of *colour tokens*; each colour token consists of a primary lightpath (i.e., wavelength number and path) and, for the SBP+NRPC problem, also a backup lightpath. Every colour token is associated with a demand whose source and destination nodes correspond to the lightpaths in the token.

A *valid solution* is a solution where each demand d is associated with at most V_d^0 tokens. A *complete solution* is a solution where each demand d is connected to exactly V_d^0 tokens; a complete solution for the example of Figure 2.1 on page 26 is shown in Figure 2.11. A *partial solution* is a valid solution that is not complete.

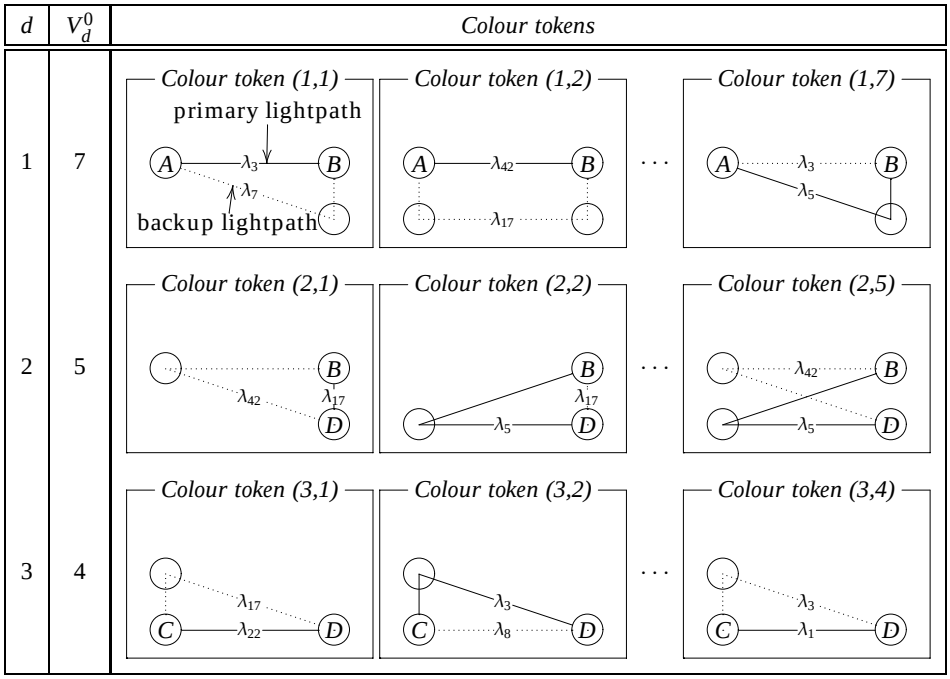


Figure 2.11: A solution consists of a set of colour tokens, each connected to a demand d

2.5.2 Elementary operations

The two heuristics use a set of elementary operations to operate on the current solution:

Allocation: A single primary lightpath and, for the SBP+NRPC design problem, a backup lightpath is allocated; that is, one token is added to the current solution.

A small number of demands are randomly chosen, and for each a small number of corresponding tokens are randomly chosen, and then for each of these, a small number of wavelengths are randomly chosen. For each such combination, the additional network cost is calculated, and finally, the cheapest one is chosen for allocation.

The number of demands, tokens and wavelength chosen is set to a predefined fraction of the available possibilities, though never less than a certain minimum number or more than a certain maximum number.

Deallocation: One or more primary lightpaths and, for the SBP+NRPC design problem, their backups are deallocated, that is, some tokens are removed from the solution. There are three flavours of deallocation:

Lightpath deallocation removes one single randomly chosen token.

Link deallocation removes all tokens with lightpaths traversing a given, randomly chosen link, freeing the duct.

Wavelength deallocation removes all tokens with primary or backup lightpaths using a given, randomly chosen wavelength.

Cost calculation: The cost of the solution is calculated. For each link, the wavelength used most times determines the needed number of fibres, which is multiplied by the fibre cost for the link before adding it to the duct cost. Finally, all link costs are added together to obtain the total cost.

All elementary operations are constructed such that they ensure that if the solution was valid before the operation, it is also valid after the operation.

2.5.3 Simulated annealing

Simulated annealing is one of the widest known metaheuristics which searches for an optimal value x in the solution space using a stochastic algorithm controlled by a decreasing temperature (Kirkpatrick et al., 1983; Aarts and Korst, 1998). It operates only on complete solutions, jumping from one solution x to a neighbouring x' in the search space if x' is better (cheaper), or at random if the temperature

is "high enough" to accept a worse solution. The initial high temperature allows the algorithm to get away from a local minimum via random jumps to worse solutions. The final low temperature ensures that when the algorithm ends, it will find a locally optimised solution, which hopefully is also a good global optimised solution.

Our instantiation of the metaheuristic for solving network design problems is shown in Figure 2.12. It is parameterised by t^{stop} which is the amount of CPU

```
SimulatedAnnealing(t^stop, γ, markovlength) =
  x ← random(SolutionSpace)          /* Choose a random solution x  */
  cost^min ← cost(x)
  x^min ← x
  T ← T_0                            /* Temperature                 */
  t ← 0                              /* Real time timer variable    */
  repeat                             /* Main loop:                  */
    for i ← 1,...,markovlength do
      x' ← DeallocateLink(x)         /* Construct randomly an x'    */
      while not complete(x) do       /* in the neighbourhood of x   */
        x' ← Allocate(x')
      Δ ← cost(x') − cost(x)         /* Compute the cost increase   */
      r ← random([0,1[)              /* Choose a random r ∈ [0,1[   */
      if Δ < 0 ∨ r < e^(−Δ/T) then   /* If x' is better or T high,  */
        x ← x'                       /* accept x' as the new solution */
        if cost(x) < cost^min then   /* If x is better than the best, */
          x^min ← x                  /* remember it                 */
          cost^min ← cost(x)         /* and its cost                */
    T ← T_0 · e^(−γ t)               /* Decrease the temperature    */
  until t ≥ t^stop
  return x^min
```

Figure 2.12: Simulated annealing core heuristic

time it should run, γ which determines how fast the temperature should decrease, and *markovlength*, which is the length of the inner loop. For each temperature, the probability of reaching a specific solution (e.g., the optimal solution) is described by a distribution function. The intention of the inner loop is that the current solution should stabilise before the temperature is lowered, changing the distribution function. In all the simulated annealing runs we have used the value 60.

Choosing temperatures T_0 and T_{stop}, we let $\gamma = (\log(T_0/T_{\text{stop}}))/t^{\text{stop}}$ such that $T_{\text{stop}} = T_0 \cdot e^{-\gamma\, t^{\text{stop}}}$. The initial x is chosen by a simple random (using a

uniform distribution) allocation of the necessary lightpaths.

Instead of deallocating lightpaths en bloc with DeallocateLink(x) we have also tried deallocating a single lightpath with DeallocateWavelength(x), but the latter turns out to give unsatisfactory results, because the high duct costs have a strong influence on the overall cost of a solution.

2.5.4 Simulated allocation

Simulated allocation is a fairly recent metaheuristic, invented by Pióro for use in optimization of telecommunication networks (Pióro, 1997). The basic simulated allocation algorithm is shown in Figure 2.13 on the next page. Contrary to simulated annealing, simulated allocation operates on partial solutions: It starts with an empty solution and then gradually constructs a complete solution, at which point it empties a lot of links at once and reiterates this construction-emptying cycle.

In the main loop a probabilistic choice is made whether to allocate a lightpath or deallocate one or more lightpaths. In order to reach a complete solution, the probability $q(x)$ for choosing allocation should be sufficiently high. Whenever a complete solution is reached, it is compared with the current best solution: If the solution is complete or the partial solution is worse than the current best (i.e., it can never improve the current best by applying only allocations), a number of links are emptied so that a different part of the solution space can be reached. The algorithm is terminated after a predetermined time t_{stop}.

The probability $q(x)$ of allocating a lightpath depends on the current partial solution x, and increases when the solution is more than 80% complete (measured in the number of tokens), to ensure that a reasonably large number of complete solutions are reached. The probability $q(x)$ as well as p^{link}, p^{bulklink} and $p^{\text{wavelength}}$ are parameters which can be adjusted to tweak the heuristic.

2.6 Empirical evaluations

Now that we have constructed the two programs shown in Figure 2.10 on page 49, the PPG and the metaheuristical optimiser (SAN or SAL), we want to evaluate their effectiveness, that is, how good the quality of their results are. We shall use two methods for evaluating the quality of the results: *benchmarking* by comparing with the results of other algorithms which are known to be optimal or to produce good results, and *known-result testing* by using the programs on problems with well-known optimal values, for example hand-constructed problems.

```
SimulatedAllocation(t^stop) =
  x ← emptysolution                          /* Start from scratch          */
  cost^min ← ∞
  t ← 0                                      /* Real time timer variable    */
  repeat                                     /* Main loop:                  */
    p ← random([0,1[)
    if p < q(x) then                         /* Choose probabilistically to */
      Allocate(x)                            /* – allocate a lightpath in x, or */
    else
      Deallocate(x)                          /* – deallocate something in x */
    if complete(x) ∧ cost(x) < cost^min then /* If x is better than the best, */
      x^min ← x                              /* remember it                 */
      cost^min ← cost(x)                     /* and its cost                */
    if complete(x) ∨ cost(x) > cost^min then /* If x is worse than the best, */
      for l ∈ links do                       /* disconnect random links in x */
        if random([0,1[) < p^bulklink then
          DeallocateLink(x, l)
  until t ≥ t^stop
  return x^min

Deallocate(x) =
  p ← random([0,1[)
  case p of                                  /* Choose probabilistically to */
  0 ≤ p < p^link:
    DeallocateLink(x)                        /* – empty a link, or          */
  p^link ≤ p < p^wavelength:
    DeallocateWavelength(x)                  /* – remove a wavelength, or   */
  p^wavelength ≤ p < 1:
    DeallocateLightpath(x)                   /* – disconnect one lightpath  */
```

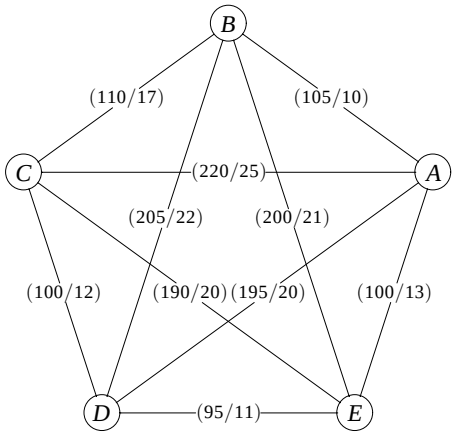
Figure 2.13: Simulated allocation core algorithm

2.6.1 Evaluating promising path generation

Benchmarking

Comparing Figures 2.6 and 2.8, we can see a way of benchmarking the PPG algorithm: As the ILP solver (GAMS / CPLEX) is able to produce optimal results (if the problem size is not too large), and the ILP generators are assumed correct, we can compare the results from the two methods—i.e., the routing and deployment—to assess the quality of the PPG algorithm.

As the arc-flow formulation is computationally very intensive, we only perform comparisons on a few small, representative networks and traffic demands. The networks and traffic demands are shown in Figures 2.14–2.16.



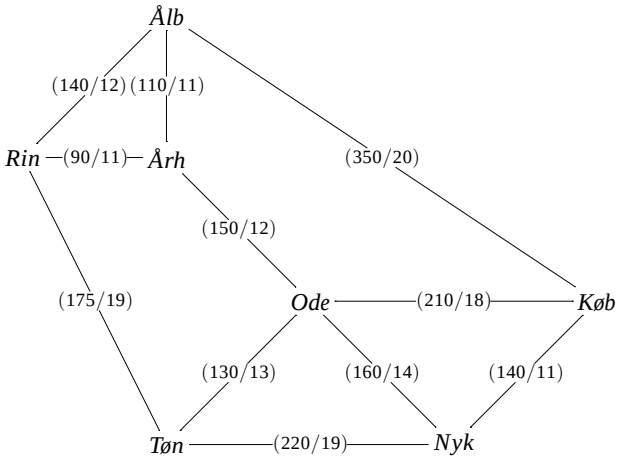
	A	B	C	D	E
A		1	7	2	5
B	0		4	6	3
C	0	0		3	5
D	0	0	0		1
E	0	0	0	0	

Figure 2.14: Example network I and traffic for assessing the quality of the PPG algorithm

The networks have the following characteristics:

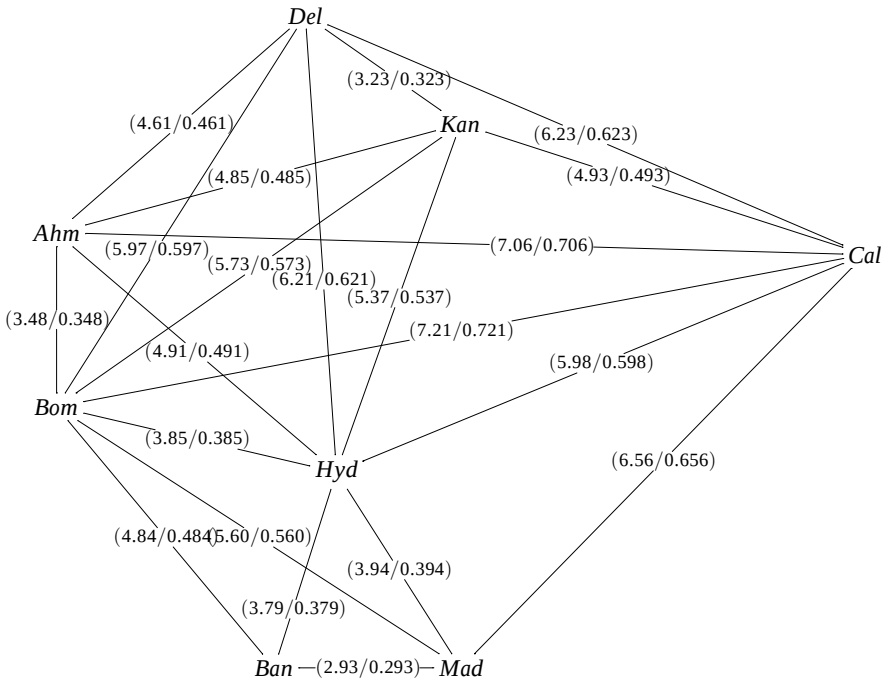
Network	*Nodes* N	*Links* L	*Connectivity* $2\frac{L-N+1}{(N-1)(N-2)}$	*Demands* D	*Demand volume* V
I	5	10	1	10	37
II	7	11	$1/3$	20	58
III	8	21	$2/3$	27	62

Connectivity is a measure of the ratio between links and nodes in a connected network, where the connectivity for a linear network is 0 and for a fully connected network is 1. For each network the PPG algorithm is run with $(k_s, k_d) = (3, 2)$



	Ålb	*Rin*	*Årh*	*Ode*	*Køb*	*Tøn*	*Nyk*
Ålb		1	5	2	6	1	2
Rin	0		2	1	1	1	0
Årh	0	0		5	8	3	3
Ode	0	0	0		6	2	2
Køb	0	0	0	0		2	4
Tøn	0	0	0	0	0		1
Nyk	0	0	0	0	0	0	

Figure 2.15: Example network II and traffic for assessing the quality of the PPG algorithm



	Del	*Kan*	*Ahm*	*Cal*	*Bom*	*Hyd*	*Ban*	*Mad*
Del		1	1	5	4	1	2	3
Kan	0		1	2	2	0	1	1
Ahm	0	0		3	3	1	1	1
Cal	0	0	0		9	1	3	5
Bom	0	0	0	0		1	3	4
Hyd	0	0	0	0	0		1	1
Ban	0	0	0	0	0	0		1
Mad	0	0	0	0	0	0	0	

Figure 2.16: Example network III and traffic for assessing the quality of the PPG algorithm

and $(8, 3)$, and $k_b = 0$, to obtain a set of promising paths. To ensure valid backup paths, k_d must be greater than 1, and the values for k_s are chosen as a compromise between a path set containing all the optimal paths and path sets of managable sizes.

Letting the number of wavelengths per fibre W be 1, 2 and 4, we find the results shown in Table 2.2 on the next page. Figures in the *value* columns are the solutions values which the given optimisation method reaches, either because it finishes or it exceeds the time limit. For arc-flow *lo bound* is the best value the solver has determined possible at timeout, while *status* indicates whether the solver stops because the solution had been proved optimal, or because the time / memory limit has been exceeded. For the *PPG* columns, '%' is the value found using the PPG, relative to the arc-flow *value* if it has been found, or otherwise the *lo bound* if that has been found.

The results indicate that PPG with $k_s = 8, k_d = 3$ produces good results in cases where the optimiser reaches a feasible solution. Unfortunately, this does not include the SBP variants—the best that can be said for them is that there is one case, SBP+NRPC for network I with 1 wavelength, that is at most 12% worse than the optimal value.

Comparing the two PPG columns it would appear that $k_s = 3, k_d = 2$ produces too few paths to allow finding a globally good solution.

Note when comparing different tasks for the same network-wavelength combinations that for values found before timeout, the ND value is a lower bound for the values of all tasks; the TRP value is a lower bound for the values of all protection tasks; and the SBP value is no greater than SBP+NRPC or SBP+FW values—which can readily be understood when considering the underlying ILP programs.

Finally we note that SBP+FW values are almost identical to SBP values, which seems to contradict results found by Baroni et al. (1999).

Known-result testing

Benchmarking against optimal ILP programs enables comparison with the optimal solution, but it has the disadvantage of limiting the problem size to unrealistically small networks. This limit can be exceeded by constructing examples where the optimal solutions can easily be seen or computed.

If we set all fibre costs to zero, the optimal result depends neither on the traffic volumes nor the wavelengths per fibre, and we find that the ND task is reduced to the classic Minimum Spanning Tree (MST) problem—to which algorithms for determining the optimal solution exist in the literature (Ahuja et al., 1993). Similarly, the TRP problem (and the other protection problems) is reduced to the classic Travelling Salesperson (TSP) problem, which current state-of-the-art

Task	*Net*	*W*	$PPG^{k_s=3}_{k_d=2}$		$PPG^{k_s=8}_{k_d=3}$		*Arc-flow*		
			value	%	*value*	%	*value*	*lo bound*	*status*
ND	I	1	1295	100	1295	100	1295		optimal
		2	916	102	894	100	894		optimal
		4	726	112	664	102		651	memout
	II	1	2283	100	2283	100	2283		optimal
		2	1587	103	1559	101		1541	timeout
		4	1242	107	1193	103		1158	timeout
	III	1	94.28	104	94.07	104	90.58		optimal
		2	62.35		62.35				memout
		4	45.92	101	45.92	101	45.61	44.85	timeout
TRP	I	1	2152	105	2088	102	2044		optimal
		2	1557		1359				memout
		4	1237		951				memout
	II	1	4038	112	3659	102	3593		optimal
		2	2885		2639				timeout
		4	2336	114	2048	100	2046		timeout
	III	1	155.09	121	129.88	101	128.56	107.10	timeout
		2	106.32	113	81.43	87	93.72	45.24	timeout
		4	81.43		71.78				timeout
PDP	I	1	2764	101	2742	101	2726		optimal
		2	1844	110	1707	102		1676	timeout
		4	1390	127	1140	104		1093	timeout
	II	1	4946	103	4783	100	4781		optimal
		2	3342	104	3228	100	3222	3218	timeout
		4	2529	148	2357	138		1712	timeout
	III	1	189.11	149	177.39	140		126.84	timeout
		2	123.13		117.33				timeout
		4	89.29	401	80.75	362		22.28	timeout
SBP	I	1	2196	156	2088	148		1411	timeout
		2	1569	239	1359	208		654	timeout
		4	1248	444	951	338		281	timeout
	II	1	4076	171	3687	154		2387	timeout
		2	2916	328	*2671	301		888	timeout
		4	2354		*2183				timeout
	III	1	155.88		132.32				timeout
		2	106.70		*114.02				timeout
		4	81.43		timeout				timeout
SBP+NRPC	I	1	2229	120	2088	112		1857	timeout
		2	1585		1359				timeout
		4	1253	346	951	263		362	timeout
	II	1	4131	136	3739	123		3044	timeout
		2	2928	213	2678	195		1372	timeout
		4	2354	377	2172	348		624	timeout
	III	1	156.74		134.45				timeout
		2	107.09		*109.88				timeout
		4	81.43		timeout				timeout
SBP+FW	I	1	2196	156	2088	148		1410	timeout
		2	1577		1359			648	timeout
		4	1248	447	951	341		279	timeout
	II	1	4076	171	3687	154		2387	timeout
		2	2916	328	2665	300		889	timeout
		4	2354		*2048				timeout
	III	1	155.88		132.32				timeout
		2	106.70		*101.88				timeout
		4	*81.43		*92.08				timeout

Table 2.2: Comparing PPG results with optimal arc-flow based values. "*" = timeout

heuristics can solve well, even for quite large instances. Furthermore, for simple, constructed examples, a solution can be found manually.

We can now test the quality of the PPG by performing the optimisation shown in Figure 2.8 on page 42. As the ILP optimiser is able to produce the best results possible with the augmented traffic produced by the PPG, and the ILP generator is assumed correct, we obtain a measure of the PPG quality by comparing the optimised deployment cost with the known optimal result.

The 30 nodes of the constructed example network we use for this test are shown in Figures 2.17 and 2.18. An optimal TSP tour and MST are shown with dotted lines, but the PPG algorithm is free to choose any of the 435 possible links that exist between node pairs, where the duct cost is identical to the euclidian distance. We let $W = 1$ and vary k_s between 1 and 20 for the ND task, setting k_d and k_b to zero, as they are intended for finding backup paths. For the TRP task we vary k_s between 1 and 10, k_d between 2 and 5, and k_b between 0 and 10. The results are shown in Table 2.3 on page 63.

We see from the results that for the ND task the PPG algorithm is able to find optimal values for $k_s \geq 5$, and even near-optimal values for $k_s = 1$. By looking at the resulting networks, shown with dashed and fully drawn lines in Figure 2.17 on the facing page, we see that the additional cost for $k_s = 1$ stems from link $I - L$.

For the TRP task we unfortunately obtain very few results due to the ILP solver exceeding the resource limits. However, from the results that are found, it seems that the PPG algorithm does not fare so well: resulting values are at least 30% worse than optimum. However, the resulting networks, shown with dashed and fully drawn lines in Figure 2.18 on page 62, look fairly “resonable” for real-world network planning.

We conclude that anyone who wants to solve a TSP problem should not use the PPG algorithm.

2.6.2 Evaluating metaheuristics

Benchmarking

Comparing Figure 2.8 on page 42 with Figure 2.10 on page 49 we see that the results in both cases depend only on the network and the augmented traffic: the effects of the PPG algorithm are—roughly speaking—removed when *comparing* the results of the two methods if we make sure to use the *same* augmented traffic. As the ILP solver is able to produce optimal results (if the problem size is not too large), and the ILP generator is assumed correct, we can assess the quality of the simulated annealing and allocation heuristics by comparing the results of the two methods shown in Figures 2.8 and 2.10.

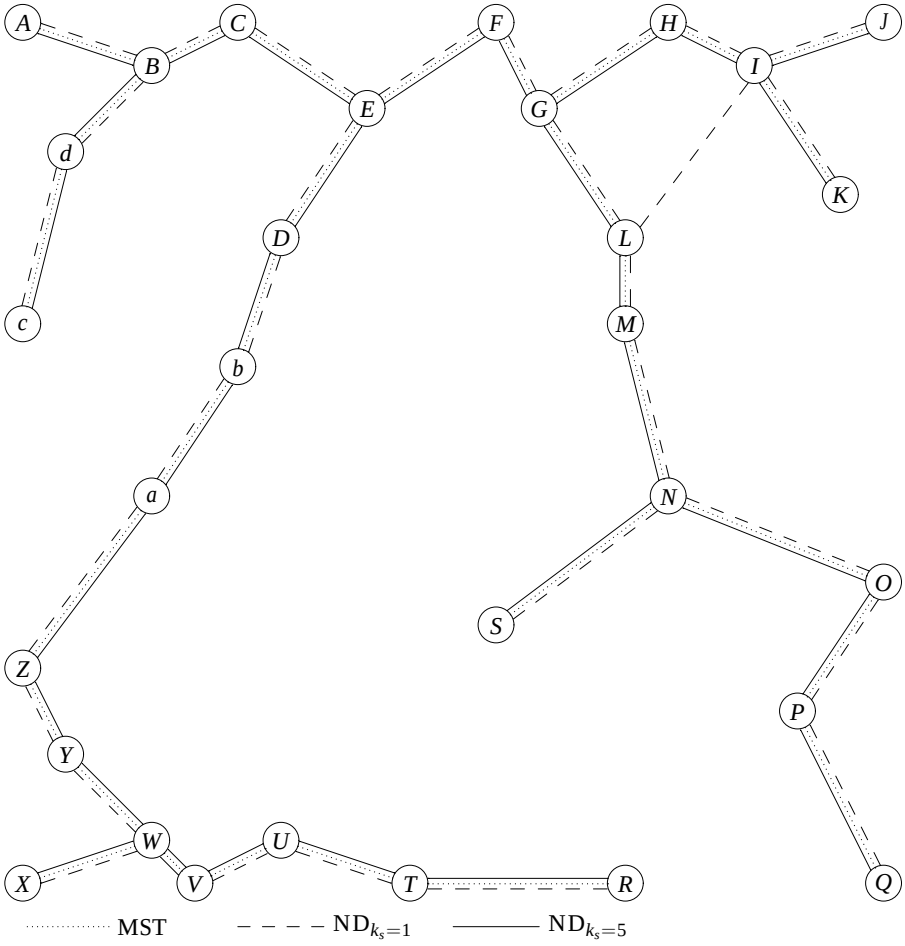


Figure 2.17: Example network for known-result testing of the ND task

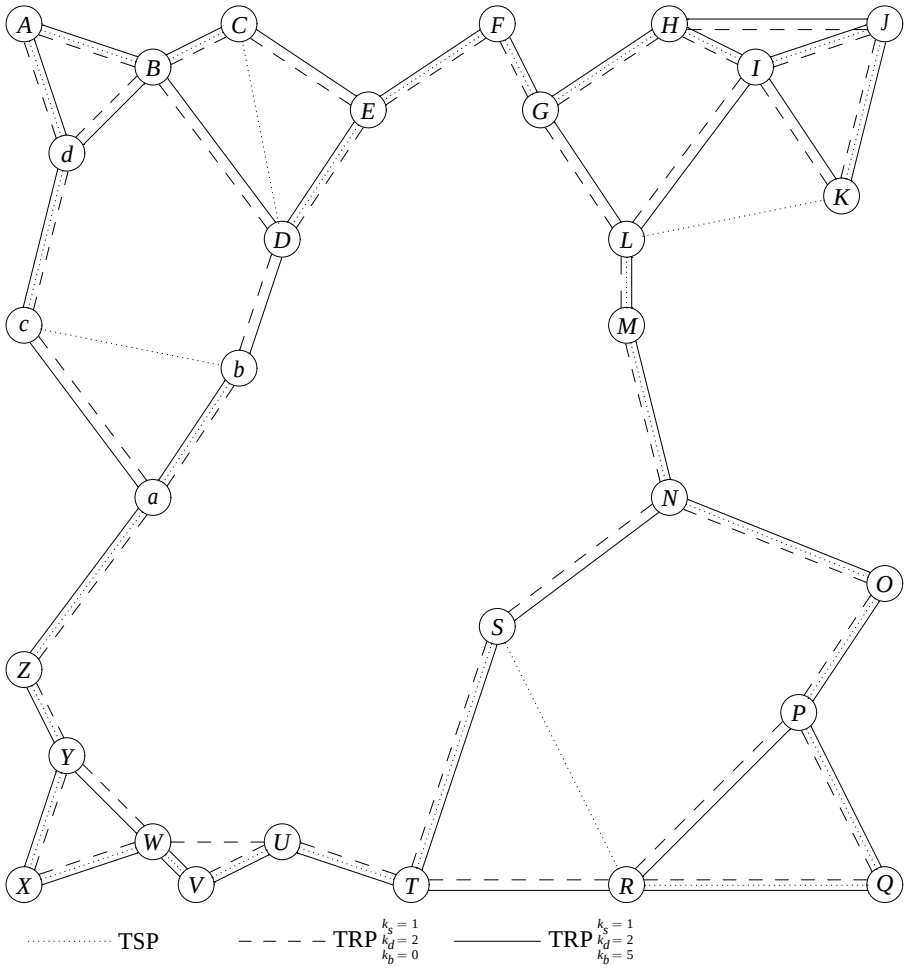


Figure 2.18: Example network for known-result testing of the TRP task

Task	k_s	k_d	k_b	*PPG* *value*	*lo bound*	%	*Known optimal*
ND	1	0	0	65.04		109	59.59
	5	0	0	59.59		100	59.59
	10	0	0	59.59		100	59.59
	20	0	0	59.59		100	59.59
TRP	1	2	0	87.14		133	65.52
			1	87.14		133	65.52
			5	85.21		130	65.52
			10	memout			65.52
		5	0	memout			65.52
			1	memout			65.52
			5	memout			65.52
			10	memout			65.52
	5	2	0	85.21		130	65.52
			1	memout			65.52
			5	memout			65.52
			10	memout			65.52
		5	0	memout			65.52
			1	memout			65.52
			5	memout			65.52
			10	memout			65.52
	10	2	0	85.21		130	65.52
			1	memout			65.52
			5	memout			65.52
			10	memout			65.52
		5	0	memout			65.52
			1	memout			65.52
			5	memout			65.52
			10	memout			65.52

Table 2.3: Comparing PPG results with known optimal values

The networks used in the experiments are two real-world examples of networks in Denmark, and a small, 5-node network:

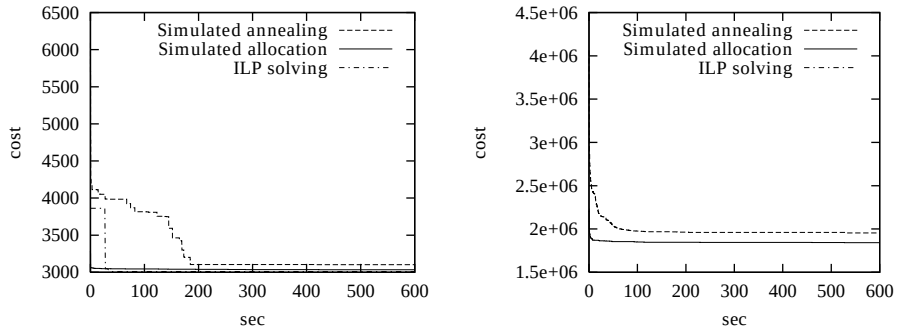
network	*nodes*	*links*	*average duct cost*	*average fibre cost*	*demands*	*demand volume*
A	5	10	152.0000	1.535000	10	71
B	23	29	213.5862	1.999310	21	844
C	73	165	31692.84	316.9284	279	1824

For each network, up to 3 mutually disjoint shortest paths and up to 5 shortest paths are generated for each pair of demand nodes. Each (task–network–wavelengths–method) configuration is run five times for 10 CPU minutes on a 440MHz HP Series 9000 Model J7000 with 1Gb RAM per optimisation process. The results are shown in Table 2.4; the status "optimal" indicates that the ILP result is proved

Task	*Net*	*W*	*Simulated allocation*		*Simulated annealing*		*ILP solving*	
			average	*std.dev.*	*average*	*std.dev.*	*result*	*status*
ND	A	4	444.00	0.00	444.60	1.79	444.00	optimal
		16	413.00	0.00	415.20	0.89	413.00	optimal
		256	404.00	0.00	404.00	0.00	—	timeout
	B	4	4384.40	1.49	4453.50	19.12	4377.68	optimal
		16	3035.02	3.26	3103.25	13.33	3010.46	timeout
		256	2605.78	1.01	2674.70	6.96	—	timeout
	C	4	2053019.09	19371.23	2089457.59	144781.07	2847760.00	timeout
		16	1898782.67	4598.40	1979889.76	97755.62	—	timeout
		256	1842618.71	4984.74	1954155.21	55768.88	—	timeout
SBP+NRPC	A	4	579.00	0.00	589.00	1.41	—	timeout
		16	531.00	0.00	537.00	1.41	—	timeout
		256	515.00	0.00	516.80	1.67	—	memout
	B	4	9614.92	7.15	9667.46	11.70	—	timeout
		16	6074.30	3.74	6083.91	4.15	—	memout
		256	4935.40	0.96	4948.83	3.11	—	memout
	C	4	3604097.35	32069.91	3705365.99	28938.54	—	timeout
		16	3341704.02	14458.96	3409685.70	48311.27	—	memout
		256	3246105.29	33566.56	3315111.54	65634.92	—	memout

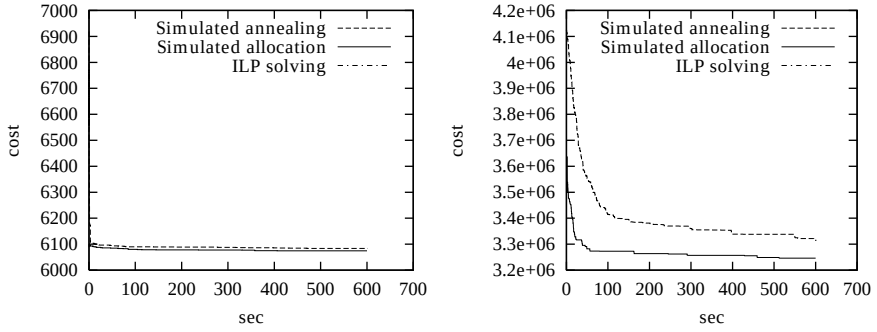
Table 2.4: Comparison of the three optimisation methods

optimal, while "timeout" or "memout" indicates that the optimisation process is terminated when the time or memory limit is exceeded, respectively. Figures 2.19 and 2.20 show examples of progress for the current best solution during optimisation as a pointwise average over the multiple runs.



Network B, 16 wavelengths Network C, 256 wavelengths

Figure 2.19: Solution cost progress during heuristical optimisation, ND



Network B, 16 wavelengths Network C, 256 wavelengths

Figure 2.20: Solution cost progress during heuristical optimisation, SBP+NRPC

The results indicate that simulated allocation is in general slightly better than simulated annealing. Furthermore, simulated allocation obtains almost optimal results in the cases where the ILP solver reaches a result within the time limit. We have also conducted experiments for 15-minute runs, and they indicate that only small improvements are gained when the heuristics are run for more than 5 minutes on these network problems. The inclusion of protection raised the cost of the net by 30–120%.

Known-result testing

Again we can use the MST and TSP problems for testing the heuristics on problems with well-known results. However, we are not able to remove the effect of the PPG algorithm, so we are in effect testing the *combination* of the PPG and heuristic algorithms. Furthermore, we compare the results with results found by Panagiotis et al. at Ericsson, who ran the commercially available tool VPItransportMaker (VPIsystems^{TM}, 2002) on the test networks.

The test networks used in this comparison have been created by Panagiotis et al.; they contain 15–1770 links and 6–60 nodes. The duct costs are proportional to the euclidian distance, and fibre costs are set to zero. There are twice as many traffic demands as nodes, each demand volume chosen at random between 0 and 20 lightpaths.

The SAL heuristic has been run with just 1 wavelength per fibre (recall that the fibre cost is zero) on a Pentium II processor with 750 MB of memory for 500 seconds. The VPI application typically took less than a second to handle each network example, according to Panagiotis.

The results are shown in Table 2.5 on the next page. Due to minor rounding errors during transfer of network data, some of the SAL values are lower than the displayed optimal values. Further, for network examples of 300 links or more, the optimal TSP tour has not been determined, but the costs of the cheapest found are shown in the table.

We note that for ND network design, SAL almost always reaches the optimal values, and in many cases produces reasonable results for the SBP+NRPC network design.

2.7 Extensions to other models

One of the useful extensions of the PPG algorithm and SAL heuristic would be to handle the network extension case as well as the greenfield case. Under the extension case some fibres would be predeployed, and using them would not require a new duct, nor any cost to deploy them. A quick and dirty first attempt could

links	*ND network design*			*SBP+NRPC network design*		
	VPI	*SAL*	*MST*	*VPI*	*SAL*	*TSP*
15	1894.0	1894.0	1894.0	2630.0	2630.0	2630.0
36	2046.7	2014.7	2014.9	2691.7	2734.4	2460.6
66	3050.9	3050.9	3050.9	3915.7	3685.9	3686.0
105	2806.7	2806.8	2806.7	4597.2	6320.6	3712.4
190	3441.2	3424.4	3424.3	4715.6	6865.5	4557.7
300	4226.6	4003.8	4003.7	6087.8	5913.3	$\leq$ 5669.6
435	4169.6	4019.0	4019.1	6129.2	6125.7	$\leq$ 5201.1
780	4909.7	4848.8	4843.4	6889.4	7949.3	$\leq$ 6322.5
1225	5439.3	5390.7	5271.1	6761.0	9764.5	$\leq$ 6586.1
1770	6118.7	6091.4	6091.7	8357.0	8587, 7	$\leq$ 7362.8

Table 2.5: Comparing results from the simulated allocation heuristic with known optima (MST and TSP costs)

be to set $C_l^{\text{duct}} = -u_l^{\text{old}} \cdot C_l^{\text{fibre}}$ where u_l^{old} is the constant number of predeployed fibres on link l. This would only work, however, if deploying new fibres does not require constructing new ducts, and if none of the links resulting from the optimisation end up with negative costs. The real way to extend the programs would be to include predeployment values explicitly. For the link-path formulation this could be done by adding nonnegative variables u_l^{new}, replacing the constraint $\delta_l = 0 \Rightarrow u_l = 0 \; \forall l$ with

$$\delta_l = 0 \Rightarrow u_l^{\text{new}} = 0 \quad \forall l \qquad \text{and} \qquad u_l \leq u_l^{\text{old}} + u_l^{\text{new}} \quad \forall l,$$

and letting the objective be to minimise $cost_{total} = \sum_l C_l^{\text{fibre}} \cdot u_l^{\text{new}} + C_l^{\text{duct}} \cdot \delta_l$. However, contrary to the quick and dirty method this is not easily implemented in the heuristics, because link deallocation must be redesigned to only deallocate an amount of lightpaths that are in *excess* of the predeployed capacity, and lightpath allocation should properly take into account that some fibres can be used "for free." The problem is to strike the right balance between greediness (using long but zero-cost detours) and efficient fibre usage (short paths using non-zero-cost links).

Finally, the general impression we get from running experiments with the network design tools is that the additional time spent assigning wavelengths during routing and link placement is not worth the slight reduction in network cost. In hindsight, it would probably have been better to split the optimisation into two phases: first routing connections, and then assigning wavelengths. On the other

hand, splitting the optimisation phase into the two separate PPG and SAL phases seems harmful, because the SAL optimiser is not able to produce new paths for a demand if it finds that all the paths supplied by the PPG would make the solution very expensive. In short, some feedback from the SAL phase to the PPG phase, possibly in an iterative fashion, would probably improve the quality of the resulting network designs.

2.8 Conclusions

In this chapter we defined some greenfield network design tasks by ILP programs, and then developed two heuristics based on simulated annealing and simulated allocation (SAL). Both heuristics require a set of precalculated paths for each traffic demand—these are supplied by a promising path generator (PPG) we constructed.

Using an ILP solver we evaluated the quality of the PPG and the heuristics; for the nominal design task the PPG seems to produce path sets of good quality, and the SAL-based heuristic compares well with results computed by ILP solvers and results computed by the commercially available VPI tool.

For protection tasks the results are less decisive, but the SAL-based heuristic produces fairly sensible although not optimal results.

Chapter 3

Optimising networks for node and link costs

The preceding chapter presented methods for greenfield network design considering just the link costs, but in reality also node costs can constitute a significant part of total network costs. In this chapter we present four studies that focus on various aspects of network optimisation including node costs.

Section 3.1 presents a way of minimising the node costs as well as the link costs in multigranular WDM networks; in Section 3.2 we propose a solution to the optical buffering problems and traffic granularity problems that exist in the nodes of all-optical networks; Section 3.3 extends the methods of the preceding chapter to include a node cost model as an integrated part of the optimisation phase; finally, in Section 3.4 we develop ILP programs for modelling ring and mesh network nodes and compare total network costs for these two architectures.

3.1 Subpath wavelength grouping

The large numbers of wavelengths which are expected in present and future backbone networks can be switched more efficiently by employing multigranular switching nodes. Instead of switching each wavelength individually, wavelengths that share common input and output links are grouped together in bands and, if sufficiently numerous, in fibres, which are switched as one unit. In this section we introduce a novel algorithm for grouping wavelengths that share a common subpath, and show that it is effective in reducing both switch port and link costs. Experiments show that this is important especially for increasing link

costs, relative to switch port costs, where the results reported by previous end-to-end grouping algorithms would deteriorate.

3.1.1 Background

The breakthrough of the Internet lead to a massive growth in data traffic during the last decade, and the response to this increased demand has been to introduce optical fibres that support higher bandwidths, adding WDM multiplexers at the backbone network nodes. With 150 wavelengths in a single fibre (Bigo et al., 1999), the OXCs must be able to switch thousands of wavelengths. While research into the construction of this kind of high capacity switch has been reported (Ryf et al., 2001; Watanabe et al., 1998; Wilfong et al., 1999), the rationale for and feasibility of switching each wavelength individually has not been clearly demonstrated.

The average number of incident links of a backbone node is typically less than ten, indicating that many wavelengths will be switched from a common input fibre to a common output fibre. This property can be exploited by grouping several wavelengths into one band, and switching this band as one unit (Gerstel et al., 2000; Harada et al., 1999; Hunter and Andonovic, 2000). With a grouping factor of W wavelengths per group, an $m \times m$ switch can potentially be reduced to an $\frac{m}{W} \times \frac{m}{W}$ switch, cf. Figure 3.1, thus reducing the number of crosspoints by a factor W^2.

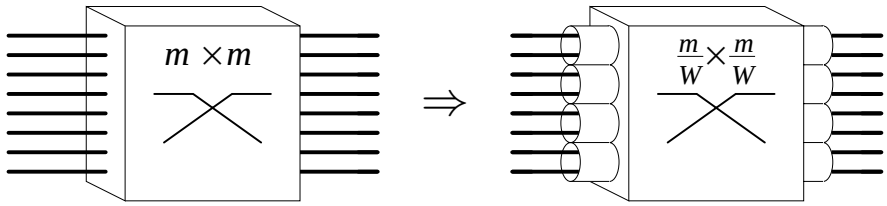


Figure 3.1: Complexity reduction potential when grouping W wavelengths together in bands

When fewer ports need to be switched, fewer optical amplifiers are needed, reducing the complexity, power consumption and multiplexer cross-talk of the switch (Ciaramella, 2000; Harada et al., 1999). This will bring down switch costs, and furthermore the reduced switching complexity will reduce the load on the network control resources. The drawback of grouping wavelengths into bands is that semi-full bands may potentially lead to wasteful fibre under-utilisation (Ciaramella, 2000).

Ciaramella (2000) considered a WRON, and performed end-to-end grouping (E2EG), grouping together 4 wavelengths with identical sources and destinations, and using a standard RWA algorithm on the resulting bands. Experiments on an Italian WRON demonstrated that for all demand volumes the number of unused wavelengths was on average about half the theoretical worst case maximum.

Yamawaku et al. (2001) also considered E2EG, but included the switch port costs as well as the link costs in evaluating the resource usage, showing for a Japanese WRON that the cost could be lowered up to 60%, depending on the port:fiber cost ratio, by grouping 6 wavelengths into each band.

Noirie et al. (2000) introduced a three-tiered multigranular model which allows for switching whole fibres and individual wavelengths, as well as bands, thus reducing the problem of unused wavelengths in the bands. Performing E2EG, this was shown to lead to a drastic reduction in the number of required switch ports, especially for nodes that switch more than 3000 wavelengths. Lee et al. (2002) considered the same problem, presenting an ILP model and a heuristic for designing multigranular networks. Furthermore, the LSP aggregation structure of GMPLS is very similar to this multigranular model, so ways of extending to support a multigranular switching model have been proposed in Internet Drafts (Dotaro et al., 2001).

The preceding papers only report results for E2EG, but Noirie et.al. have also considered prefix and suffix grouping, that is, grouping wavelengths with common sources and grouping wavelengths with common destinations along their longest common path (Noirie et al., 2001), resulting in improved results, especially for networks with low connectivity. Lingampalli and Vengalam (2002) reported good results from a similar model. Extending previous work (Glenstrup et al., 2001), we adopt their 3-level multigranular switch model and consider SWG (subpath wavelength grouping), where several wavelengths can be grouped together just a substretch of their entire path, even if their sources *and* destinations are different. This leads to a very efficient use of switch ports as well as link fibres.

3.1.2 Motivation for subpath wavelength grouping

The task solved in the present work is to assign routes and wavelengths to a set of traffic demands, each demand consisting of a source node, a destination node and a volume of requested wavelengths, in such a way that the cost of links as well as nodes is minimised.

An example of the conceptual optical switch model is shown in Figure 3.2 on the next page; Signals in fibres (thick lines) added at the node enter at the top left, and dropped signals leave it at the top right. Signals in fibres from neighbouring switches, entering at the bottom left, are either fibre-switched in

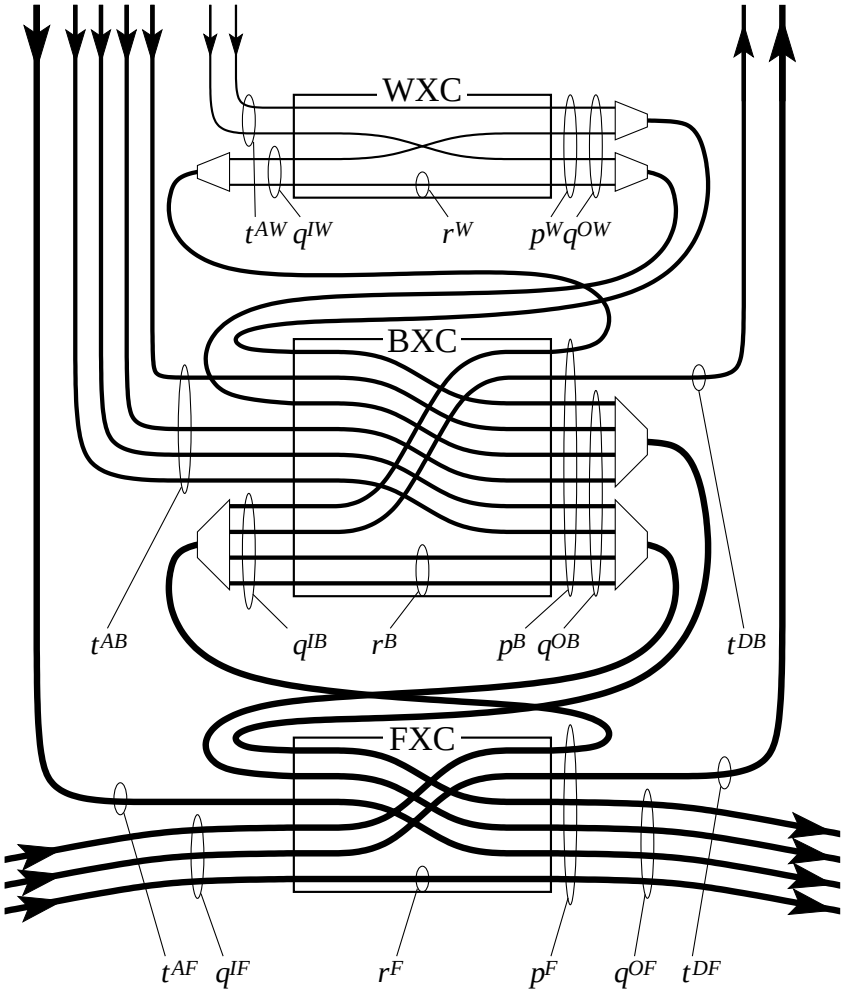


Figure 3.2: Conceptual multigranular switch model. The variables are used to indicate the sizes of various node components

the fibre cross-connect (FXC) and sent on to a neighbouring switch at the bottom right, or demultiplexed into bands and switched at the next level up, in the band cross-connect (BXC). These in turn are either band-switched or demultiplexed into wavelengths and switched in the wavelength cross-connect (WXC). Individual bands and wavelengths are added at the top of the figure. It may be necessary to introduce wavelength converters (Elmirghani and Mouftah, 2000; Harada et al., 1999), but they are not explicitly shown in this figure.

We let B denote the number of bands in each fibre, and W the number of wavelengths in each band; these numbers are identical for all fibres and switches of the network.

When deciding how to switch the connections in a multigranular WRON, several options exist. This is illustrated in Figures 3.3–3.5, for $B = 3$ bands per fibre and $W = 2$ wavelengths per band. There are 3 demands in this example: *A*–*E*: 3 wavelengths, *A*–*C*: 3 wavelengths and *D*–*E*: 2 wavelengths. The 3 wavelengths

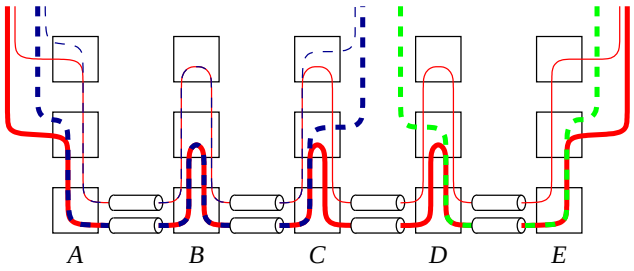


Figure 3.3: Segregated level switching

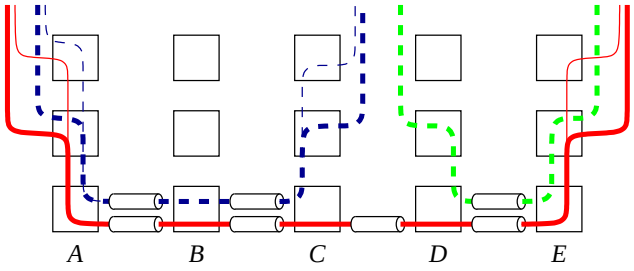


Figure 3.4: End-to-end grouping

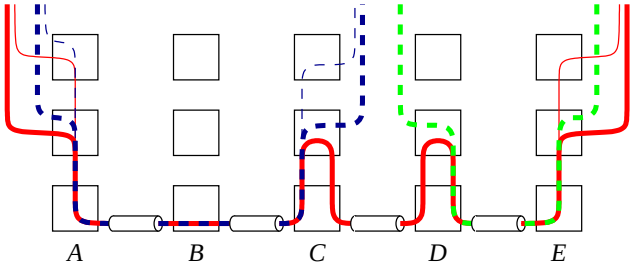


Figure 3.5: Subpath wavelength grouping

of the first two demands are added and dropped as one band (of 2 wavelengths) and one wavelength. The illustrated options are:

Segregated level switching: Full bands are band-switched using one set of fibres, and any "leftover" wavelengths are switched in another set of fibres. This strategy can be implemented using existing RWA algorithms, but the disadvantage is that several individual wavelengths might run along the same subpath, without being grouped and switched in bands, as demonstrated in Figure 3.3 on the facing page: At switch *B*, two wavelengths are demultiplexed and two bands are also demultiplexed, even though they are all switched to the same destination, and occupy in total exactly one fibre. Thus it would suffice to use just two fibre ports in switch *B* (cf. Figure 3.5), but segregated level switching uses 4 fibre ports, 4 fibre / band (de)multiplexers, 6 band ports, 2 band / wave (de)multiplexers and 4 wavelength ports.

End-to-end grouping: Only wavelengths that run on identical links along their entire path are grouped and switched together. While this strategy uses a minimal amount of switch ports, the disadvantage is a risk of too many unused wavelengths in semi-full bands, as demonstrated in Figure 3.4 on the preceding page: The bottom fibre is occupied by only two bands, one of them occupied by only one wavelength. Even if we wanted to pack demands *A*–*C* and *A*–*E* tightly along path *A*–*C* we would need two fibres, because each demand requires two bands although only three wavelengths are actually used.

Subpath wavelength grouping: The wavelengths are grouped together as much as possible along common subpaths. This strategy attempts to minimise link *and* switch port usage, the disadvantage being that routing and grouping is non-trivial. The switch port usage is slightly higher than for E2EG, as fibres

and bands must be (de)multiplexed every time a wavelength enters or leaves a group, cf. Figures 3.4 and 3.5.

As can be seen in the figures, the switch port savings arise when wavelengths are grouped together along subpaths of more than one hop. When grouping along subpaths, the key problem is to decide which wavelengths to group together, as there are many different possibilities, and grouping together in one group may reduce the switch port savings of another group (cf. Figures 3.11–3.13).

3.1.3 Routing heuristic

We develop in the following a heuristic for setting up the multigranular network switches, using a minimised number of ports and fibres.

First, we introduce some definitions. Letting *Link* denote the set of network links, we define the sets shown in Figure 3.6. We require that a subpath (π, Π)

$d \in \mathit{Demand}$	d is a demand between a node pair
$\pi \in \mathit{Path} = \mathit{List}(\mathit{Link})$	$\pi = [l_1, \ldots, l_k]$ is a list of links
$\Pi \in \mathit{PathSet} = \mathcal{P}(\mathit{Path})$	$\Pi = \{\pi_1, \ldots, \pi_k\}$ is a set of paths
$\sigma = (\pi, \Pi) \in \mathit{SubPath} = \mathit{Path} \times \mathit{PathSet}$	(π, Π) is a list of common links and the paths passing through them
$\Sigma \in \mathit{SubPathSet} = \mathcal{P}(\mathit{SubPath})$	$\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ is a set of subpaths
$\pi(i) = l_i$	denotes the ith link of path π
$\mathrm{length}(\pi) : (\mathit{Path} \to \mathit{Integer})$ $= k$, when $\pi = [l_1, \ldots, l_k]$	returns the length of path π
$\mathrm{demand}(\pi) : \mathit{Path} \to \mathit{Demand}$	returns the demand using path π
$\mathrm{usedby}(l, \Pi) : (\mathit{Link} \times \mathit{PathSet} \to \mathit{PathSet})$ $= \{\pi \in \Pi \mid l \in \pi\}$	returns the paths in Π that use link l
append(*list*, *item*) : $\mathit{List}(\alpha) \times \alpha \to \mathit{List}(\alpha)$	Append *item* to *list* and return the new list.
add(*list*, *item*) : $\mathit{List}(\alpha) \times \alpha \to \mathit{List}(\alpha)$	Adds *item* to the sorted *list* and return the new (sorted) list, removing any duplicates.

Figure 3.6: Sets, variables and functions used in the SWG heuristic

is well-formed in the sense that π is a sublist of every $\pi' \in \Pi$, that is, given $\pi' = [l_1, \ldots, l_k]$, there exist $0 \leq i, j \leq k$ such that $\pi = [l_i, \ldots, l_j]$.

If xs is a list, the notation $xs[i]$ denotes the ith element of xs.

Overall algorithm

The overall structure of the heuristic is a greedy algorithm consisting of 5 steps:

1. Determine a set Π of paths for all traffic demands.
2. Among these paths, determine common subpaths σ. For each subpath $\sigma = (\pi, \Pi)$, function $v(\pi)$ denotes the volume (number) of paths passing through π that have not yet been routed.
3. Sort these common subpaths according to value, based on their volume and number of hops, resulting in a list of subpaths $\Sigma = [\sigma_1, \sigma_2, \ldots, \sigma_k]$
4. While this list Σ is non-empty,
   (a) Take out the most valuable subpath, $\sigma_1 = (\pi, \Pi)$, from Σ. Assume it runs from node s_0 to node s_k.
   (b) From the paths in Π, determine the input port p_0, that is, all the fibres and wavelengths that the paths are expected to use when entering node s_0.
   (c) For $i \leftarrow 1, \ldots, k$, calculate from p_{i-1} the new port p_i, which is the input port at node s_i connected to the output port of node s_{i-1}. At the same time, set up the switches in nodes $s_1, \ldots, s_{k-1}$.
   (d) For all the paths $\pi' \in \Pi$, make a note that they have been partially routed along π. If π joins with some already routed subpaths of π', make the switching assignment in the joining node.
   (e) Update the list Σ of subpaths by removing parts of subpaths which overlap with π, and then re-sorting the list.
5. Set up all the switches for the adding and dropping of fibres, bands and wavelengths.

An example of how the links of the connections are grouped into subpaths can be seen in Figure 3.7 on the next page, where wavelengths are shown as black lines, common subpaths as shaded "pipes," and the numbers indicate the order in which the subpaths are allocated. Note that if subpath 4 were allocated before subpath 2, it could be widened to include subpath 6 and one wavelength from one link of subpath 2. The fact that Step 4 is iterated until Σ is empty, combined with the effect of Step 4e results in the following, illustrated by the example in Figure 3.7 on the following page:

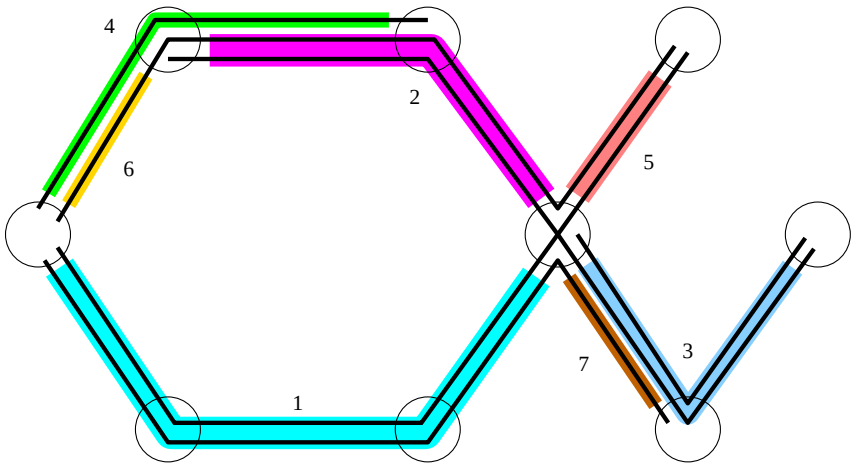


Figure 3.7: Subpath wavelength grouping in the indicated order

> The key property which is satisfied when the algorithm terminates is that each link of each path is covered by exactly one subpath.

In the following, we describe some details of the steps above.

Determining a set of paths

The paths required in Step 1 can be found in different ways, according to the scenario:

Nominal traffic routing only: Dijkstra's algorithm can be used to find shortest paths for unprotected traffic.

1+1 protected traffic routing: For 1+1 protected traffic, a 2-shortest-paths algorithm (cf. Section 2.2) can supply two paths for each demand.

There is no requirement that the paths must be the shortest paths; they could for example consist of a set of paths that avoid hot spots in the middle of the network.

Determining common subpaths

Given a path set Π, an algorithm for determining a set of common subpaths is shown in Figure 3.8 on the next page. Variable *start* is an array of starting points, indexed by the elements of Π, and Σ is the set of common subpaths.

```
subpaths(Π) =
  for π ∈ Π do                          /* consider each path π in Π        */
    Π_0 ← {}
    for i ← 1 to length(π) + 1 do       /* consider each link l_i of π      */
      if i ≤ length(π) then
        Π_i ← usedby(π[i], Π)           /* find other paths using link l_i  */
      else
        Π_i ← {}                        /* (none, if beyond final link)     */
      Π_stop ← Π_{i−1} \ Π_i            /* find common ended subpaths       */
      I ← {start[π'] | π' ∈ Π_stop}     /* find their starting points       */
      for i' ∈ I do                     /* add subpaths to Σ                */
        Σ ← add (Σ, ([l_{i'}, ..., l_{i−1}], {π ∈ Π_{i−1} | start[π] ≤ i'}))
      Π_start ← Π_i \ Π_{i−1}           /* find common started subpaths     */
      for π' ∈ Π_start do start[π'] ← i /* remember their starting link     */
  return Σ
```

Figure 3.8: Algorithm for finding common subpaths

Sorting common subpaths

The subpaths Σ should be sorted in such a way that more "valuable" subpaths are considered first. We estimate the "value" of a subpath (π, Π) to be the number of switch ports that could be saved by grouping and switching the wavelengths of the paths in Π along path π, instead of switching them individually. Given the volume $v = v(\pi)$ and length $l = \text{length}(\pi)$ of the subpath, there are $(l-1)$ switches along π, excluding the end switches; thus a first approximation to a valuation function could be $f(l,v) = (l-1) \times v$, which would allocate the longest and "widest" subpaths first. Taking the number of bands per fibre, B, and wavelengths per band, W, into account, the v wavelengths could be switched by $v \operatorname{div} (B \times W)$ fibre-switchings, $(v \bmod (B \times W)) \operatorname{div} W$ band-switchings and $v \bmod W$ wavelength-switchings. As each fibre-switching would save approximately $B \times W - 1$ ports, and each band-switching approximately $W - 1$ ports, we can refine the valuation function to

$$f(l,v) = (l-1) \times \big((v \operatorname{div} (B \times W)) \times (B \times W - 1) + ((v \bmod (B \times W)) \operatorname{div} W) \times (W - 1)\big)$$

Calculating the first output port

Figure 3.9 shows an example of how the first input port p_0 of the subpath $\sigma = (\pi, \Pi)$ is determined in Step 4b: p_0 is simply composed of the various fibres, bands and wavelengths entering s_0 which are to be routed along π. In this example, each

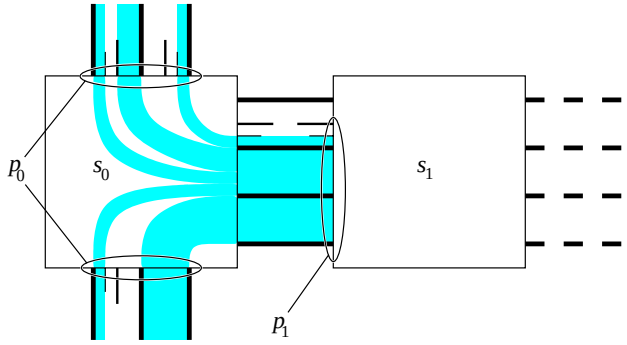


Figure 3.9: Calculating p_0, p_1, ... and setting up switches s_0, s_1, ...

of the 7 fibres connected to s_0 contains $B = 2$ bands, and each band contains $W = 2$ wavelengths. Note how the incoming wavelengths, bands and fibres are packed as tightly as possible, to enable band or fibre switching in switches $s_1, s_2, \ldots, s_{k-1}$. If some wavelengths are added at node s_0, their positions in p_0 can be chosen arbitrarily.

The algorithm for calculating the first input port at node s_0 is given in detail in Appendix A.

We take the ports from all the paths that have already been routed up to s_0, and for the remaining paths we just add a minimally-packed port of the correct volume.

Assigning subpaths

Step 4c makes sure at each intermediate switch on the subpath that the wavelengths, bands and fibres are packed tightly, making use of any unused capacity along the way, and allocating new fibres when necessary. Details are given in Appendix A.

Joining partially routed paths

When a subpath is chosen and ports are assigned, the paths that pass through this subpath are in effect assigned ports for a subset of its links. To make sure that the paths are properly routed—i.e., the correct input ports are switched to the correct output ports—we must, in Step 4d, keep track of the partial routing of the paths.

We may consider the example path π being routed in Figure 3.10 on the next page. Each time a subpath is routed, the part of the ports used at the ends of the subpath used by π must be remembered for later use in switch port setup when two allocated subpaths meet at a node.

To this end we use maps $\text{inport} : (\mathit{Path} \times \mathit{Switch}) \cup \mathit{Switch} \to \mathit{Port}$, $\text{outport} : (\mathit{Path} \times \mathit{Switch}) \cup \mathit{Switch} \to \mathit{Port}$, and the functions $\text{split} : (\mathit{Port}, \mathit{Path}) \to \mathit{Port}$ and $\text{makeswitching} : \mathit{Switch} \times \mathit{Port} \times \mathit{Port} \times \mathit{Demand} \to \mathit{Void}$:

$\text{inport}(\pi, s)$	The inport used by path π at switch s
$\text{outport}(\pi, s)$	The outport used by path π at switch s
$\text{inport}(s)$	The inport of switch s
$\text{outport}(s)$	The outport of switch s
$\text{split}(p, \pi)$	Returns the part of p which is used by π
$\text{makeswitching}(s, p_{in}, p_{out}, d)$	Set up switch s to switch the demand d from p_{in} to p_{out}

Updating Σ

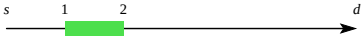
Whenever a subpath has been routed, the set Σ of subpaths must be updated in Step 4e to reflect the fact that the grouping of wavelengths into the subpath has removed them from other subpaths. In this way, some of the subpaths in Σ are replaced by shorter or more narrow subpaths. An example is shown in Figure 3.11 on the following page, where the wavelengths are shown as black lines, and three common subpaths as shaded overlapping squares. The grouping of subpath 2 will result in shorter and more narrow versions of subpaths 1 and 3. The result of this can be seen in Figures 3.12 and 3.13. This updating of Σ is accomplished by the algorithm given in detail in Appendix A; furthermore, to increase efficiency, any subpath $(\pi_1, \Pi_1) \in \Sigma$ which is completely contained in another subpath $(\pi_2, \Pi_2) \in \Sigma$, that is, $\pi_1 \subseteq \pi_2 \wedge \Pi_1 \subseteq \Pi_2$, is removed from Σ.

Setting up switches for adding and dropping

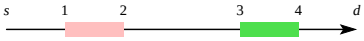
Adding and dropping fibres, bands and wavelengths at a node is very similar to finding the next output port; the details are given in Appendix A.

1. 

No switches set up yet

2. 

Make a note of $outport_1$ and $inport_2$:
$\text{outport}(\pi, 1) \leftarrow \text{split}(\text{outport}(1), \pi)$
$\text{inport}(\pi, 2) \leftarrow \text{split}(\text{inport}(2), \pi)$

3. 

Make a note of $outport_3$ and $inport_4$:
$\text{outport}(\pi, 3) \leftarrow \text{split}(\text{outport}(3), \pi)$
$\text{inport}(\pi, 4) \leftarrow \text{split}(\text{inport}(4), \pi)$

4. 

Set up switches 2 and 3,
mapping $inport_x$ to $outport_x$:
$\text{outport}(\pi, 2) \leftarrow \text{split}(\text{outport}(2), \pi)$
$\text{inport}(\pi, 3) \leftarrow \text{split}(\text{inport}(3), \pi)$
$\text{makeswitching}(2, \text{inport}(\pi, 2), \text{outport}(\pi, 2), \text{demand}(\pi))$
$\text{makeswitching}(3, \text{inport}(\pi, 3), \text{outport}(\pi, 3), \text{demand}(\pi))$

5. 

Set up switch 1, and add at switch 0:
$\text{inport}(\pi, 1) \leftarrow \text{split}(\text{inport}(1), \pi)$
$\text{makeswitching}(1, \text{inport}(\pi, 1), \text{outport}(\pi, 1), \text{demand}(\pi))$

6. 

Set up switch 4, and drop at switch 5:
$\text{outport}(\pi, 4) \leftarrow \text{split}(\text{outport}(4), \pi)$
$\text{makeswitching}(4, \text{inport}(\pi, 4), \text{outport}(\pi, 4), \text{demand}(\pi))$

Figure 3.10: Example of steps in assigning ports to a routed path

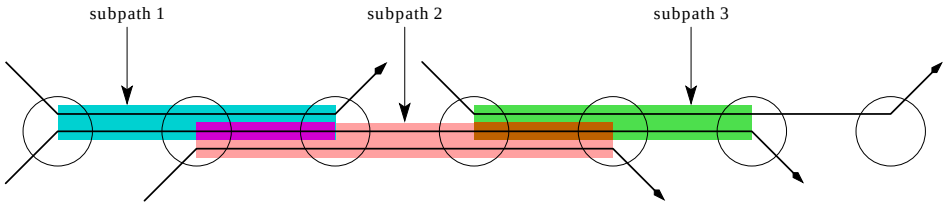


Figure 3.11: Three subpaths lapping in over each other

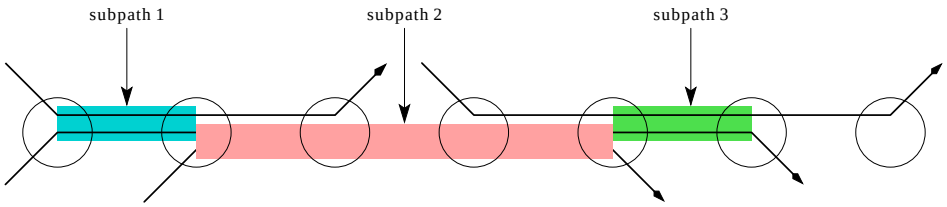


Figure 3.12: Shortening subpaths 1 and 3 after routing subpath 2

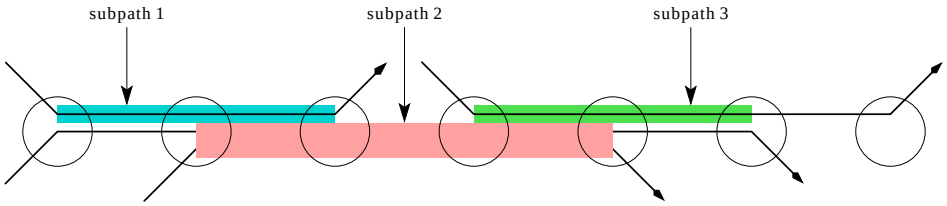


Figure 3.13: Narrowing subpaths 1 and 3 after routing subpath 2

3.1.4 Evaluations

We perform some experiments to assess various aspects of the SWG heuristic presented here. For each SWG run we can compute port usages for 3 switching models

3-level switching: This is the primary result, where both fibre, band and wavelength switches are used.

2-level switching: Disallowing fibre switching, we can convert the primary result into a 2-level switching model in which all fibre switched fibres are demultiplexed, band switched and then multiplexed again.

1-level switching: This is the conventional situation, where all fibres are demultiplexed into bands, and then into wavelengths, before they are wavelength switched and re-multiplexed again.

Note that we use the same routing for calculating the number of ports for the three models.

Counting port usage

Based on these definitions:

$f_\mathrm{i}/f_\mathrm{o}$	number of input/output fibres to the fibre OXC
$f_\mathrm{i}^\mathrm{f}/f_\mathrm{o}^\mathrm{f}$	number of fibre-switched input/output fibres
$f_\mathrm{i}^\mathrm{b}/f_\mathrm{o}^\mathrm{b}$	number of band-switched input/output fibres
$b_\mathrm{i}/b_\mathrm{o}$	number of input/output bands to the band OXC
$b_\mathrm{i}^\mathrm{b}/b_\mathrm{o}^\mathrm{b}$	number of band-switched input/output bands
$b_\mathrm{i}^\mathrm{w}/b_\mathrm{o}^\mathrm{w}$	number of wavelength-switched input/output bands
$w_\mathrm{i}/w_\mathrm{o}$	number of input/output waves to the wavelength OXC

we are able to calculate the following quantities:

$p_\mathrm{i}^\mathrm{f}/p_\mathrm{o}^\mathrm{f}$	number of input/output ports in the fibre OXC
$p_\mathrm{i}^\mathrm{b}/p_\mathrm{o}^\mathrm{b}$	number of input/output ports in the band OXC
$p_\mathrm{i}^\mathrm{w}/p_\mathrm{o}^\mathrm{w}$	number of input/output ports in the wavelength OXC

In all cases we assume that the number of fibres added and dropped at the switch is included in f_i and f_o, respectively, the number of bands added and dropped is included in b_i and b_o, respectively, and the number of wavelengths added and dropped is included in w_i and w_o, respectively.

Multigranular switching case (3-level switching) In this case, all three levels of switching are used, so we find the following fibre, band and wavelength OXC port usages $p_3^\mathrm{f}, p_3^\mathrm{b}, p_3^\mathrm{w}$:

$$\begin{array}{rcccccccl} f_\mathrm{i} + f_\mathrm{o}^\mathrm{b} & = & p_\mathrm{i}^\mathrm{f} & = & p_3^\mathrm{f} & = & p_\mathrm{o}^\mathrm{f} & = & f_\mathrm{o} + f_\mathrm{i}^\mathrm{b} \\ b_\mathrm{i} + b_\mathrm{o}^\mathrm{w} & = & p_\mathrm{i}^\mathrm{b} & = & p_3^\mathrm{b} & = & p_\mathrm{o}^\mathrm{b} & = & b_\mathrm{o} + b_\mathrm{i}^\mathrm{w} \\ w_\mathrm{i} & = & p_\mathrm{i}^\mathrm{w} & = & p_3^\mathrm{w} & = & p_\mathrm{o}^\mathrm{w} & = & w_\mathrm{o} \end{array}$$

Band- and wavelength-switching case (2-level switching) In this case, band OXCs, but no fibre OXCs, are used. Any fibres that can be fibre-switched are instead (de)multiplexed and band-switched, so we find the following band and wavelength OXC port usages $p_2^\mathrm{b}, p_2^\mathrm{w}$:

$$\begin{array}{rcccccccl} f_\mathrm{i}^\mathrm{f} \cdot n + b_\mathrm{i} + b_\mathrm{o}^\mathrm{w} & = & p_\mathrm{i}^\mathrm{b} & = & p_2^\mathrm{b} & = & p_\mathrm{o}^\mathrm{b} & = & f_\mathrm{o}^\mathrm{f} \cdot n + b_\mathrm{o} + b_\mathrm{i}^\mathrm{w} \\ w_\mathrm{i} & = & p_\mathrm{i}^\mathrm{w} & = & p_2^\mathrm{w} & = & p_\mathrm{o}^\mathrm{w} & = & w_\mathrm{o} \end{array}$$

Wavelength-switching case (1-level switching) In this case, all wavelengths are wavelength-switched, that is, the conventional case where band and fibre OXCs are not employed. All fibres and bands are (de)multiplexed before switching.

We find the following wavelength OXC port usage p_1^{w}:

$$f_{\mathrm{i}}^{\mathrm{f}} \cdot m + b_{\mathrm{i}}^{\mathrm{b}} \cdot m/n + w_{\mathrm{i}} \quad = \quad p_{\mathrm{i}}^{\mathrm{w}} \quad = \quad p_1^{\mathrm{w}} \quad = \quad p_{\mathrm{o}}^{\mathrm{w}} \quad = \quad f_{\mathrm{o}}^{\mathrm{f}} \cdot m + b_{\mathrm{o}}^{\mathrm{b}} \cdot m/n + w_{\mathrm{o}}$$

The tables presented in the following sections make use of the variables p_x^{f}, p_x^{b}, p_x^{w}, which are the numbers of fibre, band and wavelength ports, respectively, in the x-level switching model as we have just defined them, where $x = 1, 2, 3$.

SWG versus E2EG

We now turn to the question of what the merits are of SWG and E2EG. To this end we consider a Japan-like network with 47 nodes and 84 links, shown in Figure 3.14. Every link cost is set to 1, so demands are routed along fewest-hop paths. Traffic

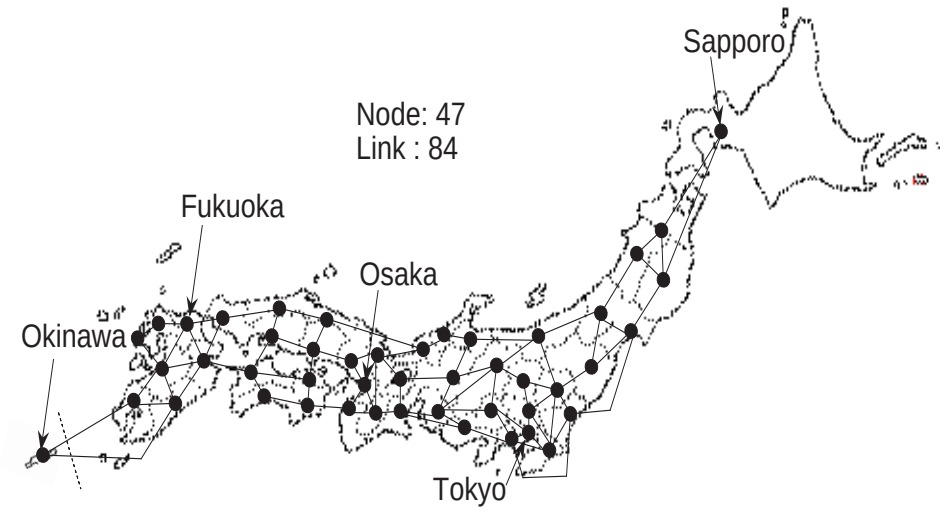


Figure 3.14: Japan-like network with 47 nodes and 84 links

is randomly generated between 2162 node pairs, based on the population sizes, with a total volume of about 12000 paths.

The results for $B \times W = 64$ and 32 wavelengths per fibre are shown in Table 3.1. We compare the SWG algorithm with results from an E2EG algorithm

		SWG													*E2EG*			
		Links			*3-level OXC Ports*				*2-level OXC Ports*				*1-level Ports*		*Links*			*Ports*
		fibres	*waves*	*utiliz.*	*fibre*	*band*	*wave*	*total*	*band*	*wave*	*total*	*reltiv*	*wave*	*reltiv*	*fibres*	*waves*	*utiliz.*	*band*
B	W	f	w	$\frac{w}{f\times B\times W}$	p_3^{f}	p_3^{b}	p_3^{w}	$\sum p_3^x$	p_2^{b}	p_2^{w}	$\sum p_2^x$	$\frac{\sum p_2^x}{\sum p_3^x}$	p_1^{w}	$\frac{p_1^{\mathrm{w}}}{\sum p_3^x}$	f	w	$\frac{w}{f\times B\times W}$	p^{b}
64	1	1018	59270	0.910	1931	37625	0	39556	71289	0	71289	1.802	71289	1.802	1022	62826	0.961	75177
32	2	1018	59270	0.910	1935	20156	2223	24314	36860	2223	39083	1.607	71289	2.932	1119	64378	0.899	41123
16	4	1018	59270	0.910	1937	11271	6377	19585	19591	6377	25968	1.326	71289	3.640	1320	65021	0.769	24311
8	8	1018	59270	0.910	1935	6658	13729	22322	10834	13729	24563	1.100	71289	3.194	1806	66538	0.576	16751
4	16	1018	59270	0.910	1926	4077	24537	30540	6201	24537	30738	1.006	71289	2.334	2986	67395	0.353	13945
2	32	1018	59270	0.910	1916	2305	30201	34422	3387	30201	33588	0.976	71289	2.071	5594	66132	0.185	13130
32	1	1938	59270	0.956	3427	31481	0	34908	71289	0	71289	2.042	71289	2.042	2007	62826	0.978	75177
16	2	1938	59270	0.956	3438	17132	2223	22793	36860	2223	39083	1.715	71289	3.128	2197	64378	0.916	41123
8	4	1938	59270	0.956	3437	9719	6377	19533	19591	6377	25968	1.329	71289	3.650	2596	65021	0.783	24311
4	8	1938	59270	0.956	3433	5882	13729	23044	10834	13729	24563	1.066	71289	3.094	3573	66538	0.582	16751
2	16	1938	59270	0.956	3412	3683	24537	31632	6201	24537	30738	0.972	71289	2.254	5930	67395	0.355	13945

Table 3.1: Fibre and port usage results for the unit link cost Japan network

performed on the same network (Yamawaku et al., 2001). The quantity w is the total number of wavelengths in all the fibres of the network, so $\frac{w}{f\times B\times W}$ is the mean fibre utilisation. It can be seen that for SWG the wavelengths are packed in such a way that the number of fibres is constant, whereas for E2EG, fibre utilisation decreases as the number of wavelengths per band increases. However, comparing the number of 2-level OXC ports with the number of E2EG ports we notice that the latter decreases as the number of wavelengths per band increases. To assess this trade-off between fibres and switch ports, we consider the ratio between the average port and link cost. Figures 3.15 and 3.16 compare total network costs for SWG and E2EG under four port:link cost ratios: 1:1, 1:10, 1:50 and 1:100. Further experiments have shown that when link costs exceed 13 times the port cost, SWG with $W = 8$ is the best grouping strategy.

Optimal grouping strategy

Given an optical network where all fibres carry the same number of wavelengths and all nodes are equipped with multigranular switches, an operator might ask what the optimal grouping strategy is, that is, how many lower-level units should be grouped together at a higher level. We propose the following

Conjecture: For a k-level multigranular network with λ wavelengths per fibre, the

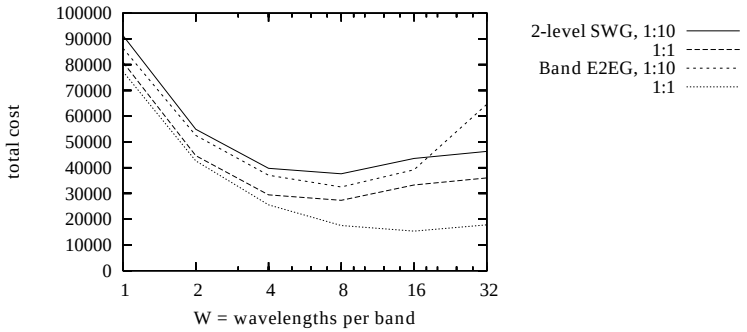


Figure 3.15: Total network cost for the Japan network for $B \times W = 64$, I

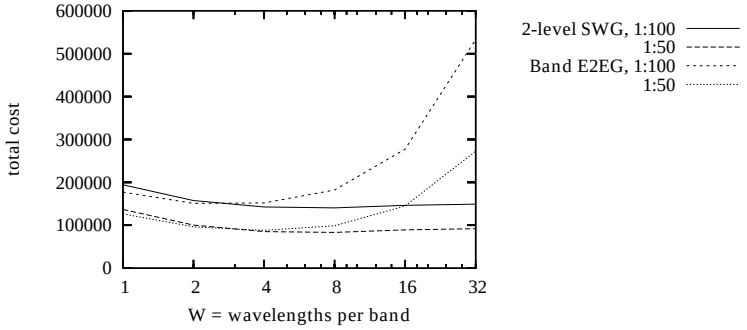


Figure 3.16: Total network cost for the Japan network for $B \times W = 64$, II

optimal grouping strategy is to group $\lambda^{\frac{1}{k-1}}$ *level-i units together at level* $i+1$.

For instance, using the 3-level switches presented here in a 64-wavelength network should group $\sqrt{64} = 8$ wavelengths in a band and 8 bands in a fibre. Using 4-level switches, on the other hand, one should group $\sqrt[3]{64} = 4$ wavelengths in each level-2 band, 4 level-2 bands in each level-3 band and 4 level-3 bands in each fibre. In cases where $\lambda^{\frac{1}{k-1}}$ is not an integer divisor of λ, one should choose divisors such that level i unit contains as close to $\lambda^{\frac{i-1}{k-1}}$ wavelengths as possible.

Table 3.2 on the facing page shows the results of running the SWG algorithm on two national networks where link distances were used to find the shortest paths:

Network	Japanese	Danish
Nodes (N)	47	12
Links (L)	84	25
Average node arity ($2L/N$)	3.6	4.2
Connectivity ($2\frac{L-N+1}{(N-1)(N-2)}$)	0.0367	0.2545
Demands	2162	44
Average demand volume	7.0	40.3
Average hop count	8.2	1.9
Average switched wavelengths	236.1	40.6

The last line indicates how many wavelengths on average are switched from one input link to another output link. In Table 3.2 on the next page the theoretically optimal values of W are shown in bold face, and in most cases for the Danish network they correspond to the optimal values found by the SWG, also shown in bold face. However, for the Japanese network, the optimal values found by SWG are smaller than the theoretical values; this is possibly due to the smaller average volume demands and the larger average number of wavelengths switched from one link to another.

3.1.5 Conclusions and future work

We introduced a novel algorithm for performing subpath wavelength grouping in a multigranular WDM network. Although simple, the algorithm achieves good results on a Japan-like network, minimising at the same time the fibre usage on the links and the switch port usage in the nodes. Compared with previous results on end-to-end wavelength grouping, fibre utilisation is higher, making SWG the best choice when link costs exceed 13 times the switch port cost.

Future research should address the added complexity which arises when limited or no wavelength conversion is used in the switches (Tripathi and Sivarajan,

		Japanese network							*Danish network*						
		Links			*3-level OXC Ports*				*Links*			*3-level OXC Ports*			
B	W	f	w	$\frac{w}{f\times B\times W}$	p_3^f	p_3^b	p_3^w	$\sum p_3^x$	f	w	$\frac{w}{f\times B\times W}$	p_3^f	p_3^b	p_3^w	$\sum p_3^x$
96	1	1338	122010	0.950	2205	57054	0	59259	49	3210	0.682	135	3888	0	4023
48	2	1338	122010	0.950	2206	30168	2998	35372	49	3210	0.682	135	1974	36	2145
32	3	1338	122010	0.950	2207	21067	5601	28875	49	3210	0.682	136	1389	75	1600
24	4	1338	122010	0.950	2209	16659	8606	**27474**	49	3210	0.682	135	1026	104	1265
16	6	1338	122010	0.950	2207	12178	14874	29259	49	3210	0.682	136	744	264	1144
12	**8**	1338	122010	0.950	2208	10325	23974	36507	49	3210	0.682	135	565	264	**964**
8	12	1338	122010	0.950	2204	7607	32238	42049	49	3210	0.682	139	433	468	1040
6	16	1338	122010	0.950	2212	6089	37022	45323	49	3210	0.682	136	336	688	1160
4	24	1338	122010	0.950	2216	4399	43230	49845	49	3210	0.682	138	272	1296	1706
3	32	1338	122010	0.950	2198	3322	44030	49550	49	3210	0.682	139	224	1392	1755
2	48	1338	122010	0.950	2211	2382	48078	52671	49	3210	0.682	135	179	2304	2618
80	1	1593	122010	0.957	2594	55774	0	58368	55	3210	0.730	150	3824	0	3974
40	2	1593	122010	0.957	2594	29480	2998	35072	55	3210	0.730	150	1942	36	2128
20	4	1593	122010	0.957	2603	16423	8606	27632	55	3210	0.730	151	1030	104	1285
16	5	1593	122010	0.957	2597	13569	11104	**27270**	55	3210	0.730	152	878	194	1224
10	**8**	1593	122010	0.957	2593	10119	23974	36686	55	3210	0.730	151	567	264	**982**
8	10	1593	122010	0.957	2593	8714	29774	41081	55	3210	0.730	152	484	424	1060
5	16	1593	122010	0.957	2594	5967	37022	45583	55	3210	0.730	151	331	688	1170
4	20	1593	122010	0.957	2594	4966	40054	47614	55	3210	0.730	151	288	1024	1463
2	40	1593	122010	0.957	2585	2719	46454	51758	55	3210	0.730	156	198	2144	2498
64	1	1973	122010	0.966	3135	51902	0	55037	65	3210	0.772	170	3408	0	3578
32	2	1973	122010	0.966	3135	27544	2998	33677	65	3210	0.772	172	1798	36	2006
16	4	1973	122010	0.966	3142	15387	8606	**27135**	65	3210	0.772	173	954	104	1231
8	**8**	1973	122010	0.966	3134	9637	23974	36745	65	3210	0.772	173	529	264	**966**
4	16	1973	122010	0.966	3129	5701	37022	45852	65	3210	0.772	175	320	688	1183
2	32	1973	122010	0.966	3124	3160	44030	50314	65	3210	0.772	172	201	1392	1765
48	1	2609	122010	0.974	4042	48174	0	52216	85	3210	0.787	204	2448	0	2652
24	2	2609	122010	0.974	4044	25728	2998	32770	85	3210	0.787	205	1278	36	1519
16	3	2609	122010	0.974	4051	18187	5601	27839	85	3210	0.787	208	941	75	1224
12	4	2609	122010	0.974	4048	14415	8606	**27069**	85	3210	0.787	205	678	104	987
8	**6**	2609	122010	0.974	4048	10714	14874	29636	85	3210	0.787	206	504	264	974
6	8	2609	122010	0.974	4040	9167	23974	37181	85	3210	0.787	205	391	264	**860**
4	12	2609	122010	0.974	4035	6847	32238	43120	85	3210	0.787	211	309	468	988
2	24	2609	122010	0.974	4062	4025	43230	51317	85	3210	0.787	207	206	1296	1709
40	1	3114	122010	0.980	4779	46694	0	51473	93	3210	0.863	214	2144	0	2358
20	2	3114	122010	0.980	4788	25120	2998	32906	93	3210	0.863	214	1102	36	1352
10	4	3114	122010	0.980	4793	14113	8606	**27512**	93	3210	0.863	214	590	104	908
8	**5**	3114	122010	0.980	4798	11857	11104	27759	93	3210	0.863	214	510	194	918
5	8	3114	122010	0.980	4782	9009	23974	37765	93	3210	0.863	214	347	264	**825**
4	10	3114	122010	0.980	4788	7850	29774	42412	93	3210	0.863	215	304	424	943
2	20	3114	122010	0.980	4782	4518	40054	49354	93	3210	0.863	214	200	1024	1438
32	1	3881	122010	0.982	5876	44926	0	50802	118	3210	0.850	245	1392	0	1637
16	2	3881	122010	0.982	5882	24152	2998	33032	118	3210	0.850	246	742	36	1024
8	**4**	3881	122010	0.982	5891	13651	8606	**28148**	118	3210	0.850	246	410	104	**760**
4	8	3881	122010	0.982	5882	8797	23974	38653	118	3210	0.850	246	257	264	767
2	16	3881	122010	0.982	5877	5291	37022	48190	118	3210	0.850	248	180	688	1116

Table 3.2: Port usage as a function of wavelength grouping sizes for two national networks

2000), and also the possibility of employing $1:n$ protection. Furthermore, it may be possible to reduce network costs even more by re-optimising the result from SWG, re-routing and re-grouping some of the most expensive wavelengths.

3.2 Synchronous optical hierarchy

In this section we introduce and describe the problems and challenges of a novel transparent optical network solution. We name this solution the *synchronous optical hierarchy* (SOH). It is a form of wavelength routed optical network where the signals occupy fractions of a wavelength through the use of timeslots in frames. Using timeslots for transparent optical networks will solve the capacity granularity problem in the wavelength routed optical network.

We define the problem of assigning wavelengths and timeslots to traffic demands by integer linear programming formulations and we solve variations over this problem by using different routing and colouring algorithms and integer linear programming. For given network, routing, traffic distribution, and parameters describing the synchronous optical hierarchy, we analytically derive the wavelength usage. The accuracy of the analytical model is found to be good when comparing with simulation results, and the model is used to derive optimal choices of timeslots under given conditions, as well as to estimate the gain from applying the synchronous optimal hierarchy.

By solving the integer linear programs for networks with static traffic assumptions we compare the efficiency of synchronous optical hierarchy networks with wavelength routed optical networks. Further, the effect of timeslot and wavelength conversion in synchronous optical hierarchy networks is also studied, as well as the effect of timeslot delays on links.

3.2.1 Background

As switching cost has become increasingly expensive (Simmons and Saleh, 1999), several ideas have been put forward to reduce the switching cost and optical-electric-optical conversion delay by applying optical switching to transit traffic (Veeraraghavan et al., 2001). Suggestions include static and dynamic WRONs (wavelength routed optical networks) (Baroni et al., 1999; Chlamtac et al., 1989; Cinkler et al., 2000; Glenstrup et al., 2000; Lee et al., 2000; Listanti and Eramo, 2000; Pióro et al., 2000; Ramaswami and Sivarajan, 1995) and optical packet switching networks (Guillemot et al., 1998; Hunter et al., 1999; Listanti and Eramo, 2000; Hunter and Andonovic, 2000). In this work we argue that SOH is a practical and efficient alternative to these structures.

The WRON, being a circuit switched all-optical network, has long been seen as the solution to drive down cost as electrical processing in large is avoided. The problem with the WRON is that the capacity granularity for connections is in wavelengths, presently up to 40 Gb/s. Traffic demands smaller than this will still occupy a whole wavelength, only making use of a fraction of the available capacity—this is the capacity granularity problem.

Optical packet switched networks could in principle solve the capacity granularity problem, because only the number of packets necessary to accommodate the traffic demands are sent through the network, but unfortunately packet switched optical networks will have to overcome major technological hurdles before they can be deployed. The problems are, among others, a lack of optical buffering and problems of optical label reading and processing (Guillemot et al., 1998; Hunter et al., 1998a,b), and as it is a packet switched technology, quality of service requirements are more difficult to fulfil.

A similar idea to the SOH was proposed by Huang et al. (2000) as time shared wavelength channels. Connections in the SOH are circuit switched like in the WRON, but since the wavelengths here consist of frames, which are divided into timeslots, the granularity of a connection is only a fraction of a wavelength, whereby the capacity granularity problem is reduced. Consequently, the technological hurdles which arise in optical packet switched networks are avoided, while the speed and much of the flexibility of optical packet switched networks is preserved.

In the SOH network wavelengths are divided into frames which are composed of timeslots, followed by gaps for synchronisation purposes, as illustrated in Figure 3.17 on the following page. One timeslot per frame is the minimum capacity unit for traffic demands, and a connection can drop in and out of different wavelengths via timeslot add/drops and timeslot cross connects. Bianco et al. (1999) and Kannan et al. (1997) have also considered timeslots, but in configurations where the wavelengths or timeslots were predetermined between specific groups of users, reducing the load flexibility for the individual end-to-end demand.

The timeslots in the SOH are synchronised, that is, all switches send a specific number of frames per second, with well-defined positions of timeslots and gaps in the frame. This characteristic and the fact that the network is circuit switched avoids the need for buffering. Further, as a connection is uniquely determined by a timeslot on a wavelength, no label reading is necessary. Thus, the buffering, label reading and processing which are the major hurdles of packet switched optical networks are entirely avoided in the SOH.

A hierarchy is naturally implemented in the SOH, thereby eliminating any granularity problems and thus leading to cost savings in networks with different traffic levels. There are several ways to introduce the hierarchy: Either a mini-

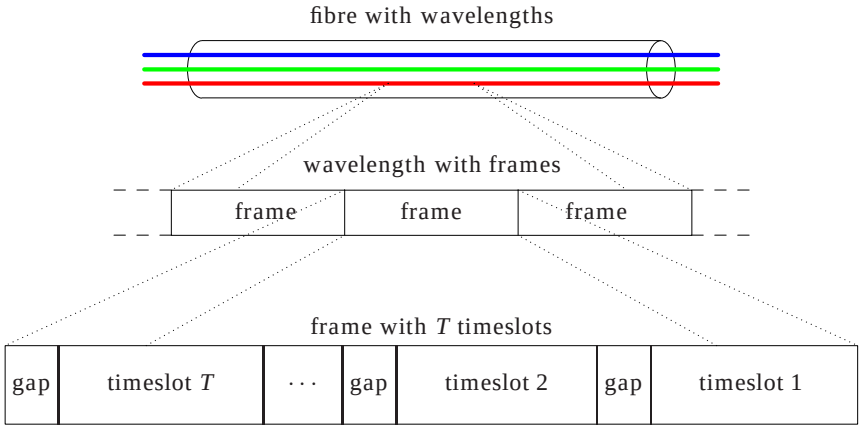


Figure 3.17: Illustration of the relations between lightpaths, frames, and timeslots.

mum timeslot length can be defined, and multiples of this timeslot length would then be available; or a minimum bit rate can be defined, and multiples of this bit rate is then available.

In this work we formulate and analyse the problems of routing and assigning timeslots and wavelengths in the SOH network. And we study the advantages of using SOH compared to WRON, measured in wavelength usage in the single fibre case, that is, the number of wavelengths needed in the maximum loaded network link to accommodate the offered amount of static traffic. Huang et al. (2000) also considered static traffic, but for the capacitated problem, measuring the percentage of allocated traffic, and only for a fixed number of timeslots per frame, either allocating or rejecting the entire amount of offered traffic between two nodes.

The problem of routing and assigning timeslots and wavelengths in the SOH network is solved both by heuristics and ILP, but in every circumstance the problem is solved sequentially, that is, routing is performed independently of the timeslot and wavelength assignment. We will comment on why this practice is only negligibly worse performing than simultaneous solution of the routing and timeslots and wavelengths assigning problem.

We define uncapacitated network design problems by ILP programs, using formulations where the paths are pre-calculated. These formulations are the well-known for WRONs presented in Chapter 2 and found in the literature (Glenstrup et al., 2000; Lee et al., 2000; Pióro et al., 2000; Ramaswami and Sivarajan, 1995), but here we extend them to include timeslots, with wavelength conversion, timeslot

conversion, and delay of timeslots. The advantage of using the formulation with pre-calculated paths instead of formulation where the paths are found by the ILP programming, like that used by Cinkler et al. (2000), is that the problem size can be kept reasonable by restricting the number of pre-calculated paths for each demand.

All our numerical experiments are performed on a Pan-European network with 19 nodes and 39 links, where we vary the number of timeslots per frame and the gap between each timeslot. Due to the novelty of the ideas presented here, the focus is on qualitative results.

3.2.2 Switching timeslots in SOH networks

When a connection in the SOH network is to be set up, a timeslot can be chosen freely at the source node, as long as the timeslot is empty. At the intermediate nodes, by default the timeslot for the next hop can be determined by that of the preceding hop. Naturally, this timeslot must not be occupied by any other connection—if this is the case, either timeslot conversion must be used, or the entire connection must use a different timeslot. Unfortunately and in contrast to wavelength conversion timeslot conversion has the drawback of delaying the signal.

Timeslot conversion in the SOH is in principle similar to wavelength conversion in the WRON. To do timeslot conversion, the signal in a timeslot must be delayed by a predetermined amount of time. Fixed, predetermined delays are easily achieved optically with fibre-optic delay lines (FDLs).

Due to the switching of packets in timeslots, they must arrive in phase at the node inputs. A simple way to synchronise the timeslots is to arrange all fibre spans connecting nodes in such a way that the time delay is equal to a multiple of the timeslot length. However, due to path variations—e.g., induced by temperature changes—this solution is not sufficiently reliable in practice, and optical synchronisers are necessary (Gambini et al., 1998; Guillemot et al., 1998; Huang et al., 2000). Furthermore, to enable exact determination of the timeslot position, gaps between timeslots are needed.

It is also conceivable to align the frames, such that timeslot 1 from all incident links enter a switch simultaneously, then timeslot 2, etc. However, frames are a virtual concept without any significance for the optical signal; switching does not become any easier for this reason, and aligning frames would cause unnecessary transmission delay.

In the case that the entire frames are not aligned, the packet on timeslot i on an input frame could be switched to a timeslot j on an output frame, where $i \neq j$. In Figure 3.18 on the next page, the timeslot synchronisation and switching is illus-

trated by an example of two input ports and two output ports with four timeslots per frame. In the figure the frames in the two input links *a* and *b* arrive out of

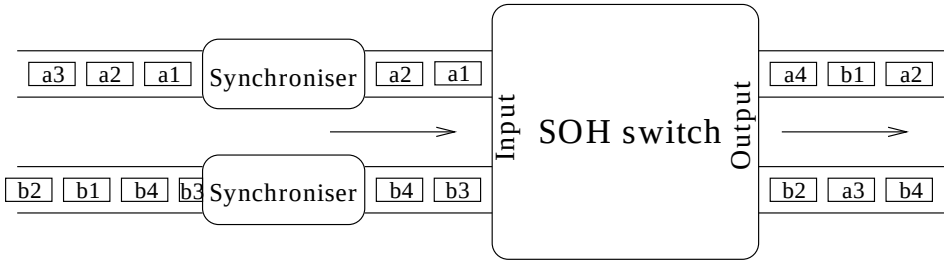


Figure 3.18: Illustration of synchronisation and timeslot switching. In this example four timeslots per frame are assumed, two input links and two output links with one wavelength per fibre, and one fibre per link.

phase, with a phase shift after synchronisation of exactly two timeslots. Thus we characterise each link by a delay, and in the following we assume that this delay, including the synchronisation, is constant and independent of the direction of the traffic on the link.

3.2.3 Mathematical formulations

In this section we give the ILP formulations of the problem of assigning timeslots and wavelengths to traffic demands.

We let a network, where each fibre holds W wavelengths and each frame consists of T timeslots, be represented by a set of L links, and the traffic by a set of D demands. The path for each demand is pre-calculated and expressed as a set of links. The routing aspect is discussed in Section 3.2.5.

The main objective is always to minimise the necessary number of wavelengths, W.

In summary, we use the following indices:

d	$\in$	$\{1,\ldots,D\}$	traffic demands
c	$\in$	$\{1,\ldots,\infty\}$	wavelengths
t	$\in$	$\{1,\ldots,T\}$	timeslots
k,l	$\in$	$\{1,\ldots,L\}$	links

We introduce constants that describe the volume of the traffic demands, the predefined paths for these demands, including an ordering relation for the path links,

and the delay:

V_d	volume of demand d (measured in timeslots)
a_{ld}	1 if link l supplies demand d, otherwise 0
p_{dkl}	1 if link k is the predecessor of link l on the path satisfying demand d, otherwise 0
Δ_l	the delay in timeslots of a timeslot transported on link l

We present five flavours of the wavelength and timeslot assignment problem for the SOH network and one wavelength assignment problem for the WRON:

SOH: Both wavelengths and timeslots must be assigned, and delays are assumed to be zero (modulo the frame length), that is, all frames at all switches are in phase. No conversion is present. Optimal result: W_{SOH}.

WRON: Only wavelengths must be assigned (there are no timeslots). No conversion is present. Optimal result: W_{WRON}.

SOH with delay: Both wavelengths and timeslots must be assigned, and delays are given for each link. No conversion is present. Optimal result: W_{delay}.

SOH with wavelength conversion: Both wavelengths and timeslots must be assigned, but wavelengths can change at each node along a path. Optimal result: $W_{\lambda\ conversion}$.

SOH with timeslot conversion: Both wavelengths and timeslots must be assigned, but timeslots can change at each node along a path. Optimal result: $W_{slot\ conversion}$.

SOH with full conversion: Both wavelengths and timeslots must be assigned, and both can change at each node along a path. Optimal result: $W_{full\ conversion}$.

In the following, we use for each flavour some decision variables to keep track of the assigned flow and wavelengths, and give a list of constraints for describing the particular problem.

ILP formulation for SOH

Variables:

x^c_{dt}	1 if demand d uses wavelength c, timeslot t, otherwise 0
y^c	1 if wavelength c is used, otherwise 0

Constraints:

$\sum_c y^c$	$=$	W		main minimisation objective
$\sum_{c,t} x^c_{dt}$	$=$	V_d	$\forall d$	satisfy all demands
$\sum_d a_{ld} x^c_{dt}$	$\leq$	1	$\forall l, c, t$	use each wavelength/timeslot on each link at most once
x^c_{dt}	$\leq$	y^c	$\forall d, c, t$	compute which wavelengths are used
x^c_{dt}, y^c	$\in$	$\{0,1\}$	$\forall d, c, t$	use only binary decision variables

Transformation of the ILP formulation from SOH to WRON

By transforming the ILP formulation for the SOH problem to an ILP formulation for the WRON problem we can show that these two problem are equivalent, whereby solving one of them solves both.

To transform the problem, introduce extra variables $y^c_t = y_c \ \forall t$, whereby the objective becomes to minimise $\sum_{ct} y^c_t = W \cdot T$. Then replace all occurrences of ct with c', replace all occurrences of c, t with c', and set $W' = W \cdot T$. Finally, replace c' with c and W' with W, to obtain the ILP for the WRON problem:
Variables:

x^c_d	1 if demand d uses wavelength c, otherwise 0
y^c	1 if wavelength c is used, otherwise 0

Constraints:

$\sum_c y^c$	$=$	W		main minimisation objective
$\sum_c x^c_d$	$=$	V_d	$\forall d$	satisfy all demands
$\sum_d a_{ld} x^c_d$	$\leq$	1	$\forall l, c$	use each wavelength on each link at most once
x^c_d	$\leq$	y^c	$\forall d, c$	compute which wavelengths are used
x^c_d, y^c	$\in$	$\{0,1\}$	$\forall d, c$	use only binary decision variables

The fact that the results for the SOH and WRON problem formulations are equivalent except for a factor T is rigorously stated and proved below, where $\lceil x \rceil$ denotes the smallest integer no smaller than x.

Theorem 1 *Given identical parameters, V_d and a_{ld} then $W_{SOH} = \left\lceil \frac{W_{WRON}}{T} \right\rceil$.*

Proof: Let the solution to the WRON problem be given, that is, let W_{WRON} and the corresponding values of x_d^c and y_c be given. As it is an uncapacitated problem, this is always possible.

Without loss of generality we assume that $y_c = 0$ for all $c > W_{WRON}$; this also implies that $x_d^c = 0$ for all $c > W_{WRON}$ and $y_c = 1$ for all $c \leq W_{WRON}$. Let $W' = \lceil W_{WRON}/T \rceil$, and for $c' = 1, \ldots, W'$ and $t = 1, \ldots, T$ let $x_{dt}^{c'} = x_d^c$, where $c = t + T(c'-1)$, and let $y^{c'} = \max_t\{y_{t+T(c'-1)}\}$. This implies that $y^{c'} = 1$ for all $c' \leq W'$. We find that constraints of the SOH formulation are indeed fulfilled with the variables W', $x_{dt}^{c'}$, and $y^{c'}$:

$$\begin{array}{rcll} \sum_{c'} y^{c'} & = & W' & \\ \sum_{t,c'} x_{dt}^{c'} = \sum_{t=1}^{T} \sum_{c'=1}^{W'} x_{d,t+T(c'-1)} = \sum_{c=1}^{T \cdot W'} x_d^c = \sum_c x_d^c & = & V_d & \forall d \\ \sum_d a_{ld} x_{dt}^{c'} = \sum_d a_{ld} x_d^c & \leq & 1 & \forall l, c', t \\ x_{dt}^{c'} = x_d^c \leq y_c & \leq & y^{c'} & \forall d, c', t \end{array}$$

To see that this solution is optimal, i.e. $W' = W_{SOH}$, suppose there exists a better solution $W'' = \lceil W_{WRON}/T \rceil - a$, where a is a positive integer. Again, without loss of generality assume that $y^{c'} = 0$ for $c' > W''$ and thus $x_{dt}^{c'} = 0$. Setting $y^{c'} = y_c$ and $x_{dt}^{c'} = x_d^c$ for $c = 1, \ldots, W'' \cdot T$, where $c' = \lceil c/T \rceil$, we find that

$$\begin{array}{rcll} \sum_{c'} x_{dt}^{c'} = \sum_c x_d^c & = & V_d & \forall d \\ \sum_d a_{ld} x_{dt}^{c'} = \sum_d a_{ld} x_d^c & \leq & 1 & \forall l, c' \\ x_{dt}^{c'} = x_d^c & \leq & y_c = y^{c'} & \forall d, c' \end{array}$$

but $\sum_c y_c = \sum_{c'} T \cdot y^{c'} = T \sum_{c'} y^{c'} = T \cdot W'' = T(\lceil W_{WRON}/T \rceil - a) < W_{WRON}$, which contradicts the optimality of W_{WRON}. Therefore $W_{SOH} = \lceil W_{WRON}/T \rceil$. □

A trivial addition to Theorem 1 is that the wavelength usage for the SOH network in case of timeslot and wavelength conversion, denoted by $W_{\textit{full conversion}}$, is related to wavelength usage in the WRON with conversion, $W_{\textit{WRON, full conversion}}$, by the equivalent equation:

$$W_{\textit{full conversion}} = \left\lceil \frac{W_{\textit{WRON, full conversion}}}{T} \right\rceil$$

It can be seen that a gain in wavelength usage by using wavelength converters and timeslot converters in the SOH network leads to a gain in the corresponding WRON. The opposite is not true in general.

ILP formulation for SOH with delay

Variables:

x^c_{dtl}	1 if demand d uses wavelength c in timeslot t on link l, otherwise 0
y_c	1 if wavelength c is used, otherwise 0

Constraints:

$\sum_c y_c$	$=$	W		main minimisation objective
$\sum_{t,c} x^c_{dtl}$	$=$	$a_{ld}\, V_d$	$\forall d, l$	satisfy all demands
$\sum_d a_{ld} x^c_{dtl}$	$\leq$	1	$\forall l, c, t$	use each wavelength/timeslot on each link at most once
x^c_{dtl}	$\leq$	y_c	$\forall l, d, c, t$	compute which wavelengths are used
x^c_{dtk}	$=$	$x^c_{dt'l}$	$\forall d, c, l, t,$ $k: p_{dkl}=1,$ $t': t'=(t+\Delta_k) \bmod T$	change timeslot position of demand use the predecessor link to link l t' is the new timeslot after link k
x^c_{dtl}, y_c	$\in$	$\{0,1\}$	$\forall l, d, c, t$	use only binary decision variables

In this formulation we must keep track of the timeslot used on each link of a path, which requires an extra index l on the flow variable x. We then add a constraint ensuring that if link k is a predecessor of link l on the path of demand d, the timeslots are cyclically delayed Δ_k timeslots when going from k to l. Note that the modulus operator used differs slightly from the traditional definition. Traditionally, if the left argument is a multiple of the right argument the results is zero, but we have: $k \cdot n \bmod n = n, \forall k \in \mathbb{N}$.

ILP formulation for SOH with wavelength conversion

Variables:

x_{dt}	number of wavelengths used by demand d in timeslot t

Constraints:

$\sum_t x_{dt}$	$=$	V_d	$\forall d$	satisfy all demands
$\sum_d a_{ld} x_{dt}$	$\leq$	W	$\forall l, t$	compute number of wavelengths used for each timeslot on each link
x_{dt}	$\in$	$\mathbb{N}_0$	$\forall d, t$	decision variables are non-negative integers

With wavelength conversion the wavelength index c on the x-variable is not needed.

ILP formulation for SOH with timeslot conversion

Variables:

x_d^c	number of timeslots used by demand d on wavelength c
y_c	1 if wavelength c is used, otherwise 0

Constraints:

$\sum_c y_c$	$=$	W		main minimisation objective
$\sum_c x_d^c$	$=$	V_d	$\forall d$	satisfy all demands
$\sum_d a_{ld} x_d^c$	$\leq$	T	$\forall l, c$	use at most T timeslots on each wavelength of each link
$x_d^c \geq 1$	$\Rightarrow$	$y_c = 1$	$\forall d, c$	compute which wavelengths are used
x_d^c	$\in$	$\mathbb{N}_0$	$\forall d, c$	number of timeslots is a non-negative integer
y_c	$\in$	$\{0, 1\}$	$\forall c$	wavelength usage is a binary decision variable

In this formulation we need not keep track of precisely which timeslots are used for each demand, only the total number for each wavelength. As $x_d^c \leq T$, the implication constraint is simply implemented in the ILP program as $x_d^c \leq T \cdot y_c$.

ILP formulation for SOH with full conversion

Constraints:

$\sum_d a_{ld} V_d$	$\leq$	$T \cdot W$	$\forall l$	each link can carry T timeslots on each wavelength

With full conversion the number of needed wavelengths is determined by the links loaded with the highest volume of traffic.

Relationships between optimal values of different formulations

We mention without rigorous proof that given the same traffic demands V_d and routing a_{ld}, the following relationships between the optimal values of W found

by the various ILP formulations hold:

$$\begin{aligned} W_{SOH} &\geq W_{\lambda\ conversion} \geq W_{full\ conversion} \\ W_{SOH} &\geq W_{slot\ conversion} \geq W_{full\ conversion} \\ W_{delay} &\geq W_{slot\ conversion} \geq W_{full\ conversion} \\ W_{SOH} &= \lceil W_{WRON}/T \rceil \end{aligned}$$

In each case, the solution whose value is represented on the left side of one of the inequalities will also be a feasible solution to the problem represented on the right side.

3.2.4 Gain from using timeslots

In this section we analytically show the possible savings from using timeslots.

Theorem 2 *Assume a zero gap, wavelength conversion, and that the traffic demand volume, Λ_d, between each node-pair in granularity of wavelengths (as opposed to V_d which was given in units of timeslots) is chosen from the uniform distribution from 0 to K, where K is a positive integer. Assume further that one link carries a significantly higher number of paths than all other links in the network. Then the number of expected wavelengths is*

$$\lambda = \left(\frac{K}{2} + \frac{1}{2T}\right) D,$$

where T is the number of timeslots per frame, and D is the number of demands on the link, which carries the maximum number of paths.

Proof: The average usage of timeslots for one demand is

$$\sum_{k=1}^{KT} P(k-1 < \Lambda_d T < k)\, k = \sum_{k=1}^{KT} \frac{k}{KT} = \frac{(KT)^2 + KT}{2KT} = \frac{KT}{2} + \frac{1}{2}$$

Where $P(\cdot)$ denotes the probability. The average use of timeslots on a link with d demands is found by multiplication by d:

$$\left(\frac{KT}{2} + \frac{1}{2}\right) d$$

The average use of wavelength is then given by the smallest larger integer after division by T:

$$\left(\frac{K}{2} + \frac{1}{2T}\right) d$$

Since one link carries a significantly higher number of paths than all other links in the network then the wavelength usage, λ, is determined by that link, so

$$\lambda = \left(\frac{K}{2} + \frac{1}{2T}\right) D$$

□

With uniformly distributed demand sizes it is seen that the higher the number of timeslots per frame the larger are the savings, but the savings are never more than 50%. The higher the upper bound is in the uniform distribution (determined by K), the smaller are the relative savings by using timeslots.

When we assume a non-zero gap and non-integer parameters, the proof for the average wavelength usage becomes more technical ($\lfloor x \rfloor$ denotes the greatest integer no greater than x):

Theorem 3 *Assume a gap length fraction g, where $0 \leq g < 1/T$ and T is the number of timeslots per frame. Assume wavelength conversion, and that the traffic demand volume between each node-pair in granularity of wavelengths is chosen from the uniform distribution from 0 to K, where K is positive. Assume further that one link carries a significantly higher number of paths than all other links in the network. Then the number of expected wavelengths is*

$$\lambda = \frac{1-Tg}{KT^2}\left(\frac{1}{2}\left\lfloor\frac{KT}{1-Tg}\right\rfloor^2 + \frac{1}{2}\left\lfloor\frac{KT}{1-Tg}\right\rfloor + \left\lceil\frac{KT}{1-Tg}\right\rceil\left(\frac{KT}{1-Tg} - \left\lfloor\frac{KT}{1-Tg}\right\rfloor\right)\right) D,$$

where D is the number of demands on the link, which carries the maximum number of paths.

Proof: One traffic demand can fill up to $\left\lceil\frac{KT}{1-Tg}\right\rceil$ timeslots. With Λ_d being the demand volume of demand d in granularity of wavelengths, the average usage of timeslots for one demand becomes

$$\begin{aligned}
&\Sigma_{k=1}^{\left\lceil\frac{KT}{1-Tg}\right\rceil} P\left(k-1 < \tfrac{\Lambda_d T}{1-Tg} < k\right) k \\
= &\left(\Sigma_{k=1}^{\left\lfloor\frac{KT}{1-Tg}\right\rfloor} P\left(k-1 < \tfrac{\Lambda_d T}{1-Tg} < k\right) k\right) + \left\lceil\tfrac{KT}{1-Tg}\right\rceil P\left(\left\lceil\tfrac{KT}{1-Tg}\right\rceil - 1 < \tfrac{\Lambda_d T}{1-Tg} < \left\lceil\tfrac{KT}{1-Tg}\right\rceil\right) \\
= &\left(\Sigma_{k=1}^{\left\lfloor\frac{KT}{1-Tg}\right\rfloor} \tfrac{1-Tg}{KT} k\right) + \left\lceil\tfrac{KT}{1-Tg}\right\rceil \tfrac{1-Tg}{KT}\left(\tfrac{KT}{1-Tg} - \left\lfloor\tfrac{KT}{1-Tg}\right\rfloor\right)
\end{aligned}$$

Following the same steps as in the proof for Theorem 2 on the facing page, we get the stated expected wavelength usage. □

3.2.5 Results and discussion

In this section we first look at the idealised SOH network were the gap is zero and verify the principles of Theorem 2 on page 100. An ensemble of non-uniform traffic matrices are used for statistical purposes. Secondly, we analyse an example in depth with the traffic generated from the gravitational model. The network used in all cases is the The Pan-European network used in the OPEN project (Chbat et al., 1998) consisting of 19 nodes and 39 links, shown in Figure 3.19.

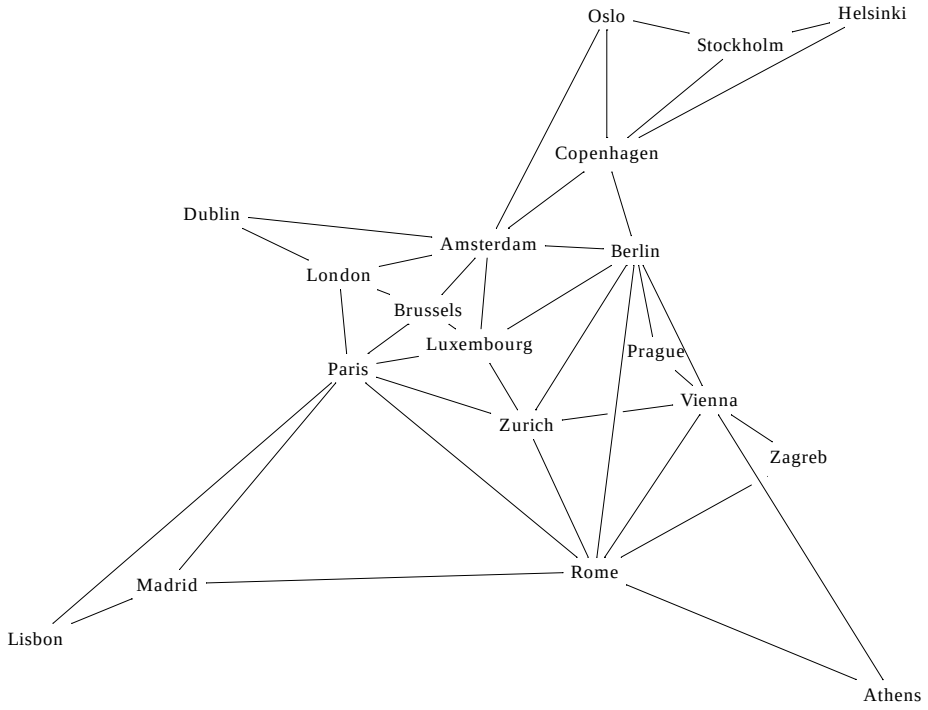


Figure 3.19: The Pan-European network consisting of 19 nodes and 39 links.

Generation of traffic

We give the traffic demand matrix in units of a given bit rate or in units of wavelengths, which corresponds to some bit rate.

Consider the structure of a frame, shown in Figure 3.20. We let f denote the frame length in time, g the gap length in time, t the timeslot length in time, and T the number of timeslots in a frame. As each frame consists of T timeslots and T gaps, we have that $f = (t+g)T$, that is, $t = f/T - g$.

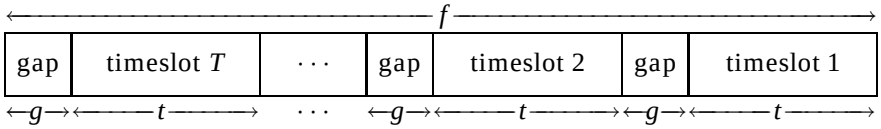


Figure 3.20: Frames are composed of T timeslots and gaps

Fixing f to one time unit, the required number of timeslots, V_d, for satisfying a specific traffic demand, Λ_d, varies only with T and g. We disallow a fractional number of timeslots—any such values are rounded up to the nearest integer, so the required number of timeslots is $V_d = \lceil \Lambda_d / t \rceil = \lceil \Lambda_d / (1/T - g) \rceil$. It is seen that if $g = 1/T$ then no number of timeslots can accommodate the traffic demand. In this case the gaps between the timeslots in the frame fill up the entire frame, such that the timeslot length has to be zero which, of course, is useless.

Routing

We distinguish between what we call simple shortest path routing and optimised shortest path routing. Both routing methods refer to complete routing between all node-pairs in the networks between which a traffic demand exists. We define simple shortest path routing as routing using shortest paths without consideration of congestion of the links. Optimised shortest path routing is defined as routing making use of only shortest paths, but where the paths are chosen such that the link with the highest traffic load carries as little traffic as possible. The metric used here is number of hops. For simplicity only one path for each demand is considered.

The method for finding the optimised shortest path routing used here is similar to the one used by Baroni and Bayvel (1997). It is a near optimal method, so we cannot guarantee that the most congested link carries as little traffic as possible. The path allocation is performed as follows: For each node-pair an alternative shortest path substitutes the one previously assigned when the number of channels used on the most loaded link with the alternative path is lower. The process is repeated until no further improvement can be made. Although allowing longer paths than the shortest path between a node-pair could lower the the number of channels used on the most loaded link, we do not allow this, since the overall

capacity usage would be increased. For simplicity we route all demands between two nodes on the same path. The routing depends on the traffic demands; when the demands change, rerouting must be performed.

Verification and consequence of analytical result

To confirm the validity of Theorem 2 on page 100 in Section 3.2.4 we have made simulations for simple shortest path routing and optimised shortest path routing. We have studied traffic from different symmetric uniform distributions in the interval $[0;K[$ for $K = 0.5, 1, 2, 4$ with up to 32 timeslots per frame. To have data material for statistical use, up to 10^5 different traffic matrices have been simulated for each K in case of simple shortest path routing and up to 10^3 different traffic matrices have been simulated for each K in case of optimised shortest path routing.

We have plotted the results showing the average wavelength usage as function of number of timeslots per frame. Strictly speaking, the average wavelength usage is the average load in units of wavelengths for the link carrying the most traffic, that is, the average wavelength usage in case of wavelength converters. Theorem 1 on page 96 gives the connection between wavelength usage in SOH networks and WRONs. To find the wavelength usage it therefore suffices to set up the algorithms for the WRON. According to several studies (Baroni and Bayvel, 1997; Fenger et al., 2000), the difference in wavelength usage with and without wavelength converters for uniform traffic (i.e., identical traffic demand for all node-pairs) is negligible. The values of the elements in the traffic matrix are uniformly distributed, so the entire matrix is in general non-uniform. By using the same methods as Fenger et al. (2000) we still find that the difference in wavelength usage with and without wavelength converters is negligible. Therefore the results are valid both with and without wavelength converters. This is important since the use of timeslot converters would cause unnecessary delay of signal.

Since the difference in wavelength usage is almost independent of the presence of wavelength converters, the difference in wavelength usage between the sequential solution to the routing and timeslot and wavelength problem used here, and the optimal results based on simultaneously solving the routing and timeslot and wavelength problem, is negligible.

The results for simple shortest path routing are shown in Figure 3.21 on the facing page. We have included error bars on the result for the 95% confidence interval. The confidence intervals have the length of four times the estimated spread of the mean value of wavelength usage under the assumption that the error is Gaussian distributed.

It can be seen that in all cases the wavelength usage determined by the theory is almost identical to or slightly less than the average of the numerical results. This

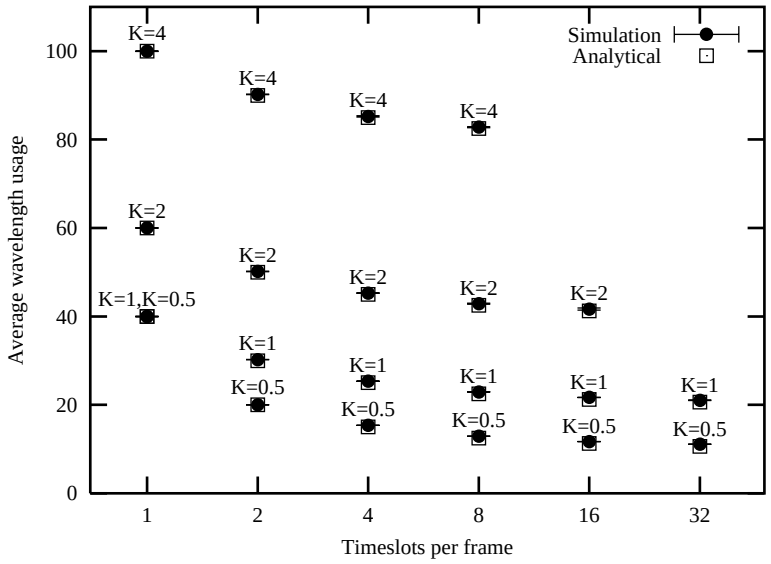


Figure 3.21: Analytically predicted wavelength usage compared to simulation in case of simple shortest path routing. The 95% confidence interval is shown for the simulation results. Traffic is symmetric, uniformly distributed in the interval $[0; K[$. The gap is zero.

is due to the fact that our assumption that the wavelength usage λ is determined by one link only will not hold, since not only the most congested link in terms of demands can have the largest traffic volume. The result obtained theoretically will therefore be a lower bound for the average wavelength usage for the SOH network in case of simple shortest path routing.

Optimised shortest path routing results are shown in Figure 3.22. Although

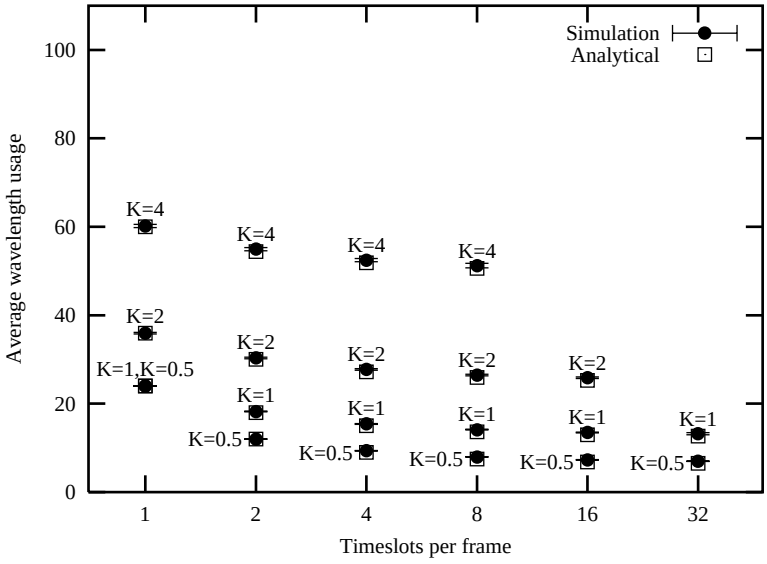


Figure 3.22: Analytically predicted wavelength usage compared to simulation in case of optimised shortest path routing. The 95% confidence interval is shown for the simulation results. Traffic is symmetric, uniformly distributed in the interval $[0; K[$. The gap is zero.

the number of paths on the maximum loaded link varied from traffic matrix to traffic matrix, we did not show the error bars for the analytical results, since the spread here was smaller, about 50% or less. Both the simulation result and the theoretical results for the wavelength usage varies with the traffic matrix. In the figure the variance of the estimated mean value is only shown for the simulation results, since variance of the estimated mean value of the theoretical result were always significantly lower. All the results for average wavelength usage for simulation are higher than or very close to the theoretically expected wavelength usage. Optimised shortest path routing levels out congestion, whereby

more links are candidates to be the most congested link in terms of largest traffic volume, which should raise the average wavelength usage compared to the theory more than that in the case of simple shortest path routing. For the results with optimised shortest path routing the theoretical wavelength usage varies as the traffic varies since the routing takes the traffic into account, whereby the number of paths on the most congested link in terms of number of paths varies.

By comparison of the Figures 3.21 and 3.22 it is seen that the average wavelength usage for simple shortest path routing for all points is about 50% higher than with optimised shortest path routing.

When the gap is zero it is seen from Theorem 2 on page 100 that the absolute gain in wavelength usage from using SOH is almost independent of K except for variations in D, when K is an integer. However, the relative gain increases and as seen from Figure 3.21 on page 105 and Figure 3.22 on the facing page for large T ($T \gtrsim 8$) the wavelength usage is approximately proportional to K. From Theorem 3 on page 101 it is also seen that the wavelength usage becomes approximately $\frac{1}{2}KD$ for large T.

With small demand volumes then in case of only one timeslot per frame, many demands will not efficiently fill up a timeslot, so the utilisation will be low, and dividing the frames into several would raise the utilisation and lower the wavelength usage as we have seen. If typical demand volumes are several wavelengths then for one timeslot per frame the relative number of inefficiently filled timeslots would have been smaller. In conclusion, the smaller the demands are, the larger the gain is from using SOH.

Theorem 3 on page 101 gives the behaviour of the wavelength usage for given parameters, K, T, D. In Figure 3.23 on the next page we have plotted the behaviour for different values of T. For each T the wavelength usage is a growing function and approaches infinity as g approaches $1/T$. In terms of wavelength usage the most efficient choice of T for $T \in \{1, 2, 4, 8\}$ and $K = 1$ for a given g can be derived from Figure 3.23 on the following page. For given g then the T representing the lowest curve is the optimal choice. For instance, if $g \lesssim 0.02$ then $T = 8$ yields the lowest number of wavelengths, and if $0.02 \lesssim g \lesssim 0.08$ then $T = 4$ is the optimal choice.

The size of the gap is given in fractions of frame length, and is therefore—disregarding technicalities—determined by the length of a frame. The length of a frame is important for delay considerations, that is, there is a trade off between delay and utilisation.

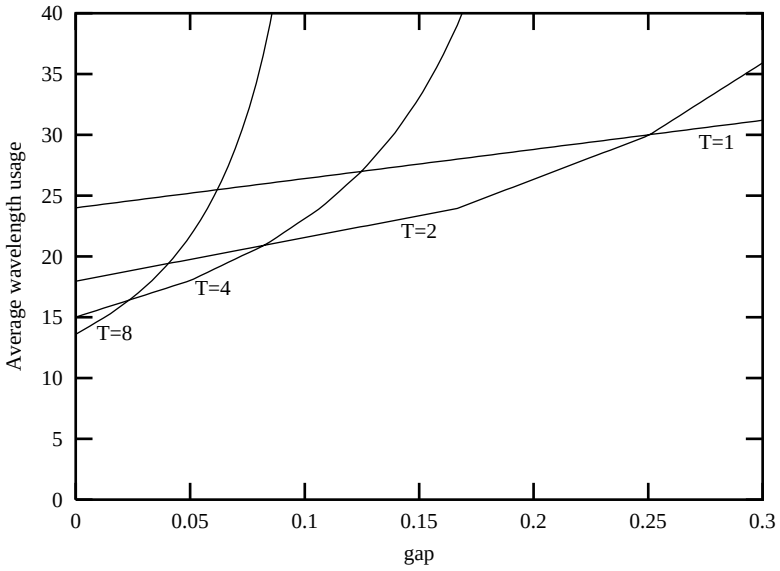


Figure 3.23: Wavelength usage versus gap size for different numbers of timeslots per frame according to Theorem 3. Traffic is symmetric, uniformly distributed in the interval $[0; 1[$.

Solution to ILP problems

We perform some experiments with the ILP programs to determine whether there are any significant differences between the various formulations. The Pan-European network shown in Figure 3.19 on page 102 is the basis for the experiments, where the traffic demand matrix is generated from the gravitational model. The point of using this network and traffic is not to guarantee realistic traffic, which we cannot. Rather, this is a known network with a traffic demand matrix created by a well known model. Any network with any traffic demand matrix could be used as long as they are not too computationally demanding.

The traffic to be routed is static and symmetric, with each demand Λ_d between two nodes measured in units of wavelength, that is, in fractions of frame length. To generate the traffic we have used the gravitational model: the traffic demand between two cities is the product of their population sizes, scaled by an appropriate constant (10^{-15}). The traffic demand matrix for the network in Figure 3.19 on page 102 is shown in Table 3.3. As an example, the traffic demand between Helsinki and Berlin is 0.43 wavelengths. This number is found by multiplying the number of citizens in Finland with the number of citizens in Germany and scaled: $5.2\,10^6 \cdot 8.3\,10^7 \cdot 10^{-15} = 0.43$. If one wavelength corresponds to 40 Gb/s then the traffic between Finland and Germany corresponds to 21.2 Gb/s in each direction. The traffic ranges from 0.0017 to 5 wavelengths.

	Helsinki	Stockholm	Oslo	Copenhagen	Berlin	Amsterdam	Dublin	Prague	Luxembourg	Brussels	London	Vienna	Zurich	Paris	Zagreb	Athens	Rome	Madrid	Lisbon
Helsinki	0	0.046	0.023	0.028	0.43	0.083	0.02	0.053	0.0023	0.053	0.31	0.042	0.038	0.31	0.022	0.055	0.3	0.21	0.052
Stockholm	0.046	0	0.04	0.048	0.74	0.14	0.034	0.091	0.0039	0.091	0.53	0.072	0.065	0.53	0.038	0.094	0.51	0.36	0.089
Oslo	0.023	0.04	0	0.024	0.37	0.072	0.017	0.046	0.002	0.046	0.27	0.037	0.033	0.27	0.02	0.048	0.26	0.18	0.045
Copenhagen	0.028	0.048	0.024	0	0.44	0.086	0.021	0.055	0.0024	0.055	0.32	0.044	0.039	0.32	0.023	0.057	0.31	0.21	0.054
Berlin	0.43	0.74	0.37	0.44	0	1.3	0.32	0.85	0.037	0.85	5	0.68	0.6	4.9	0.36	0.88	4.8	3.3	0.84
Amsterdam	0.083	0.14	0.072	0.086	1.3	0	0.061	0.16	0.0071	0.16	0.95	0.13	0.12	0.95	0.069	0.17	0.92	0.64	0.16
Dublin	0.02	0.034	0.017	0.021	0.32	0.061	0	0.039	0.0017	0.039	0.23	0.031	0.028	0.23	0.017	0.041	0.22	0.15	0.039
Prague	0.053	0.091	0.046	0.055	0.85	0.16	0.039	0	0.0045	0.11	0.61	0.084	0.075	0.61	0.044	0.11	0.59	0.41	0.1
Luxembourg	0.0023	0.0039	0.002	0.0024	0.037	0.0071	0.0017	0.0045	0	0.0045	0.026	0.0036	0.0032	0.026	0.0019	0.0047	0.026	0.018	0.0045
Brussels	0.053	0.091	0.046	0.055	0.85	0.16	0.039	0.11	0.0045	0	0.61	0.084	0.075	0.61	0.044	0.11	0.59	0.41	0.1
London	0.31	0.53	0.27	0.32	5	0.95	0.23	0.61	0.026	0.61	0	0.49	0.43	3.6	0.26	0.63	3.4	2.4	0.6
Vienna	0.042	0.072	0.037	0.044	0.68	0.13	0.031	0.084	0.0036	0.084	0.49	0	0.059	0.49	0.035	0.087	0.47	0.33	0.082
Zurich	0.038	0.065	0.033	0.039	0.6	0.12	0.028	0.075	0.0032	0.075	0.43	0.059	0	0.43	0.032	0.077	0.42	0.29	0.073
Paris	0.31	0.53	0.27	0.32	4.9	0.95	0.23	0.61	0.026	0.61	3.6	0.49	0.43	0	0.26	0.63	3.4	2.4	0.6
Zagreb	0.022	0.038	0.02	0.023	0.36	0.069	0.017	0.044	0.0019	0.044	0.26	0.035	0.032	0.26	0	0.046	0.25	0.17	0.044
Athens	0.055	0.094	0.048	0.057	0.88	0.17	0.041	0.11	0.0047	0.11	0.63	0.087	0.077	0.63	0.046	0	0.61	0.43	0.11
Rome	0.3	0.51	0.26	0.31	4.8	0.92	0.22	0.59	0.026	0.59	3.4	0.47	0.42	3.4	0.25	0.61	0	2.3	0.58
Madrid	0.21	0.36	0.18	0.21	3.3	0.64	0.15	0.41	0.018	0.41	2.4	0.33	0.29	2.4	0.17	0.43	2.3	0	0.4
Lisbon	0.052	0.089	0.045	0.054	0.84	0.16	0.039	0.1	0.0045	0.1	0.6	0.082	0.073	0.6	0.044	0.11	0.58	0.4	0

Table 3.3: Traffic demand matrix in units of wavelengths.

Optimisation results The ILP programs have been solved optimally on a 1GB 440MHz HP J7000 using GAMS and CPLEX 7.1.

We have varied the number of timeslots for each of the five flavours of the SOH ILP problems presented in Section 3.2.3, and we have varied the size of the gap for the SOH ILP. We used optimised shortest path routing, which has been performed separately for each number of timeslots, as the routing depends on the traffic.

In Table 3.4 we give the wavelength usage as a function of the number of timeslots in a frame with a gap of 0.01 frame length between each timeslot. The wavelength usage is given for the five different flavours of the SOH ILP problems presented in Section 3.2.3. We have 1 to 32 timeslots per frame, where 1 timeslot per frame corresponds to WRON except for a gap on each frame, since a timeslot is followed by a gap.

timeslots	W_{SOH}	W_{delay}	$W_{\lambda\ conversion}$	$W_{slot\ conversion}$	$W_{full\ conversion}$
1	24	24	24	24	24
2	16	16	16	16	16
4	14	14	14	14	14
8	14	14	14	14	14
16	15	*out of memory*	15	15	15
32	18	*out of memory*	18	18	18

Table 3.4: Number of necessary wavelengths for the Pan-European network (Figure 3.19) as function of the number of timeslots with traffic demands ten time those given in Table 3.3. The gap between timeslots is 0.01 frame length.

It can be seen that wavelength conversion, timeslot conversion or both does not reduce the wavelength usage, which confirms and extends previous results on the RWA problem for unprotected WDM networks (Baroni et al., 1999; Fenger et al., 2000; Ramaswami and Sivarajan, 1995). Furthermore, the introduction of delays, such that connections must use different timeslots on different links of their routes, does not raise the wavelength usage. The delay for each link was chosen randomly and uniformly from one to the number of timeslots on a frame. For 16 and 32 timeslots per frame it was not possible to solve the linear problem due to too high memory usage; this could possibly be avoided by using the independent set formulation described in Section 2.4.3 on page 47. The minimum wavelength usage is reached with 4–8 timeslots per frame. Compared to 1 timeslot per frame, which corresponds to a WRON with frames, at least 62.5% fewer wavelengths are needed with SOH at the most favourable number of timeslots.

3.2.6 Conclusions

We have described the concept of SOH networks, which we see as an optical solution to alleviate the capacity granularity problem.

Further, we have defined the problem mathematically by ILP formulations in the case of static traffic for various cases of conversion capabilities for wavelengths and timeslots and the presence of delay of timeslots on links.

From the ILP formulation it has been possible to reduce the problem of allocating demands in the SOH network to the simpler problem of allocating demands in the WRON, which we have shown by manipulating the ILP formulation.

Given routing, traffic distribution, and parameters describing the SOH we have analytically derived an approximate result for the expected wavelength usage. We have compared simulation results to the analytical results and found very good agreement for the Pan-European network.

We found that simple shortest path routing has a wavelength usage which is about 50% higher than that of optimised shortest path routing.

From the analytical result we have illustrated a method to determine the optimal number of timeslots per frame as function of the gap size, which has practical importance for determining the tradeoff between delay and utilisation. Further, we find the gain from using timeslots as function of the gap size. For optimised shortest path routing the wavelength usage is up to 100% higher without the use of timeslots.

By using ILP we have compared wavelength usage for WRONs and SOH networks with different numbers of timeslots per frame. The network used was the Pan-European network and the traffic was generated from the gravitational model. We have studied 5 different setups of the SOH: no conversion, conversion for wavelengths, timeslots, and both, and signal delay. All setups required the same number of wavelengths for the studied number of timeslots per frame, implying that delay and conversion capabilities of neither wavelengths nor timeslots have any significant effect. The required number of wavelengths was significantly reduced with SOH; in this case more than 40% fewer wavelengths were needed.

We find that it is advantageous to use SOH in all-optical networks; the gain is particularly great when the typical traffic demand between source and destination is less than a few wavelengths.

3.3 Integrated node and link optimisation

In this section we present a general node model for all-optical networks and develop a method for designing a greenfield network in an optimised fashion, given

a static traffic demand matrix. The method, an extension of the heuristic based on simulated allocation developed in Section 2.5, optimises the design to minimise the overall cost of optical nodes and fibre links. We investigate the feasibility of using this method on networks of realistic sizes, and compare integrated optimisation of ducts, fibres and nodes with staged optimisation where the network is optimised first with respect to ducts and fibres, whereupon the node deployment is calculated.

In this work we present ILP formulations of the two uncapacitated greenfield AON design problems for static traffic demands: ND where traffic demands are only assigned primary paths, and SBP+NRPC where additionally each primary path is assigned a backup path, cf. the list in Section 2.1 on page 27. To facilitate handling larger networks, we also extend the SAL heuristic developed in Section 2.5.

3.3.1 Background

We use the same link cost model as previously, the sum of the fibre cost and an opening cost for duct construction. As Kershenbaum et al. (1991) note, this can also model any node costs that are proportional to the number of fibre ports. Hjelme and Andersen (1999) found that the overall network costs scale with different network parameters—hop length, link length, node degree—according to whether duct, fibre or node costs dominate. However, *which* factor is dominating may not necessarily be clear before the network is designed, so instead of optimising specifically for reducing hop length, link length or node degree, we *include* the node cost in the optimisation process. Furthermore, we also add a quadratic fibre port factor, which allows modelling more precisely switches whose costs are proportional to the size of the switch matrix. Cinkler et al. (2000) also consider node costs for several different node components (OXCs, DXCs, OADMs etc.) in their optimisation, but that is done indirectly by separating the network into W layers—one for each wavelength—and converting the nodes into small subgraphs according to their functionality and incorporating them in the layered network. This layering method has also been used by Sun et al. (2001), but the layers increase dramatically the complexity of the problem, and the method does not in itself allow for the quadratic cost factor. Furthermore, although the nodes are included in the optimisation, the *kind* of node must be selected by the user before optimisation starts.

As our main concern is the cost of network elements rather than wavelength assignment, we assume in this work that all nodes are equipped with wavelength converters.

3.3.2 Network model

We consider a greenfield scenario in which a telecommunication operator has decided where to locate the network nodes, and has estimated the costs of constructing ducts and deploying fibres between the nodes. Again, we refer to these estimates as (potential) links; it is up to the optimisation process to select which links should actually be constructed, and how many fibres to deploy in each duct. Although we do not require estimates for *all* node pairs, the intention is that there should be sufficiently many potential links for the optimisation process to choose between.

The traffic which the network must be designed to support is given as a static set of traffic demands (a traffic matrix), where each demand consists of a source node, a destination node and a traffic volume, measured in units of wavelengths, which is to be realised.

Each fibre supports a fixed number of wavelengths, W, and links can contain as many fibres as is necessary. When fibres are deployed along a link, there must also exist a duct, so the cost model for a link l is given by

$$\text{cost}_l(f) = \begin{cases} C_l^{\text{fibre}} f + C_l^{\text{duct}}, & f > 0 \\ 0, & f = 0, \end{cases}$$

where f is the number of fibres deployed, and C^{fibre} is the estimated cost of deploying one fibre in the duct which has an estimated construction cost of C^{duct}. Typically, C^{duct} is much larger than C^{fibre}.

Each node is equipped with an OXC which we assume has no bounds on the number of fibre ports it can handle, and is able to perform strictly non-blocking switching between all input and output ports. The node cost for any node n in the network is modelled as a second order polynomial:

$$\text{cost}_n(f) = \begin{cases} C^{\text{switching}} f^2 + C^{\text{port}} f + C^{\text{core}}, & f > 0 \\ 0, & f = 0, \end{cases}$$

where f is the number of fibre ports required in the node. This allows for modelling switches where the cost is proportional to the switching matrix size. We require $C^{\text{switching}} \geq 0$ and $C^{\text{port}} \geq 0$, as the node cost function must be increasing for positive f to avoid "getting stuck" in non-global minima during the optimisation process; similar requirements exist in other work on network design optimisation (Kershenbaum et al., 1991).

3.3.3 Network design optimisation

The objective of the network design task is to satisfy all the traffic demands, minimising the total network cost given as the sum of the link and node costs. For

the ND design task just primary paths are required, while each primary path in the SBP+NRPC design task must also be assigned a backup path. We note that as the number of fibres and switch ports in not bounded, we are considering an *uncapacitated* network design problem.

We consider two ways of optimising the deployment of the network components—the nodes and links:

Staged optimisation, in which we first optimise the links, that is, decide where to construct ducts and how many fibres to deploy in them. This is followed by a second stage where the node equipment is optimised, given the fixed fibre deployment. For the quadratic node model described in Section 3.3.2, this simply corresponds to postprocessing, calculating the cost of each node, but one could envisage other node models in which switching functionality or port restrictions would leave room for further optimisation by choosing between a set of node architectures.

Integrated optimisation, in which both nodes and links are optimised in the same stage, allowing node costs to influence the layout of the ducts and fibres.

The best choice between these two optimisation strategies is a main focus point of this work, and it depends among other things on the relative sizes of the node and link costs and the traffic volumes, as well as the effect of the added complexity of integrated optimisation, which risks slowing down the optimisation process.

Path set generation

We augment the PPG developed in Section 2.3 by transforming the network before it enters the core PPG algorithm, replacing each original node by one input and one output node, connected by a link which can model the appropriate cost model for the node. An example is shown in Figure 3.24 on the facing page. Each bidirectional link is modelled in the transformed network by two unidirectional links, each going from the output node of its source to the input node of its destination. Each traffic demand enters at its input source node and terminates at its output destination node. This way, the algorithm of Figure 2.5 on page 32 can directly be reused.

Optimisation by integer linear programs

We present in the following an ILP formulation of the network design problems we are considering in this work.

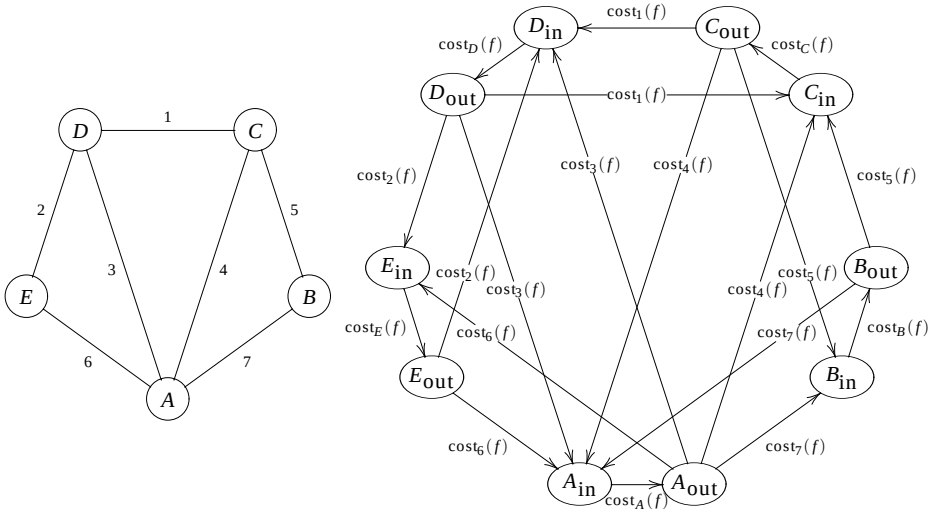


Figure 3.24: Example of how node costs can be modelled by link costs in a transformed network

Approximating the polynomial node cost model. As the node cost function $\text{cost}_n(f)$ is not linear, we cannot express it directly in the ILP framework. Instead we approximate the non-constant part, $\widehat{\text{cost}}_n(f) = C^{\text{switching}} f^2 + C^{\text{port}} f$ with the piecewise linear function $\widehat{\text{cost}}_n^{\text{ILP}}(f) = \max_i\{K_i f + C_i^0\}$ as sketched in Figure 3.25 on the following page. Note that $C^{\text{core}} + \widehat{\text{cost}}_n^{\text{ILP}} \leq \text{cost}_n$ because $C^{\text{switching}} \geq 0$.

The coefficients for the set of linear functions $\{K_i f + C_i^0 \mid i = 1, \ldots, I\}$ are determined by selecting I points $\big(f_0, \widehat{\text{cost}}_n(f_0)\big), \ldots, \big(f_I, \widehat{\text{cost}}_n(f_I)\big)$ on the graph for $\widehat{\text{cost}}_n(f)$ and finding the tangential slope $K_i = 2C^{\text{switching}} f_i + C^{\text{port}}$ by differentiation. The cost offset C_i^0 is determined by solving $\big(\widehat{\text{cost}}_n(f_i) - C_i^0\big) \;/\; \big(f_i - 0\big) = K_i$, which yields $C_i^0 = -C^{\text{switching}} f_i^2$. Naturally, the accuracy of the approximation is better when then number of points is large, but for our purposes we expect that $I = 12$ points are sufficient if we make sure that the largest point is greater than the number of ports expected at any node.

The ILP program in the arc-flow formulation. Next we shall describe the components of the ILP program for the arc-flow formulation of the ND network design task. In this formulation, the input and output flow at each node, including

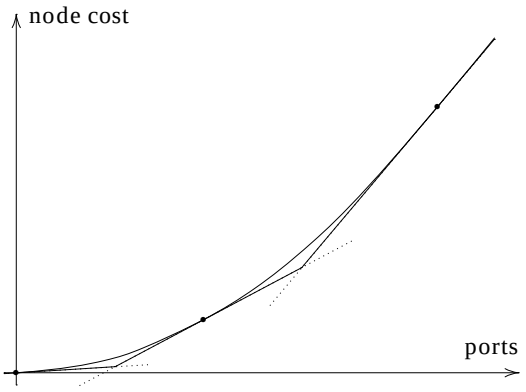


Figure 3.25: Approximating the node cost function by a piecewise linear function

locally added and dropped flow, must be balanced. Additionally, we use the K_i and C_i^0 parameters described in the preceding section to calculate the node costs, based on the total input and output flow at the node. In the constraints the quantification "$m : (nm) \in \mathcal{E}$" denotes "all nodes m for which there exists an edge (nm) from node n to node m in $\mathcal{E}$."

Indices:

$d \in \{1, \ldots, D\}$	Traffic demands
$m, n \in \{1, \ldots, N\}$	Network nodes
$(nm) \in \mathcal{E} \subseteq \{1, \ldots, N\}^2$	Directed edges
$\{n, m\} \in \mathcal{L} \subseteq \mathcal{P}(\{1, \ldots, N\})$	Undirected links
$i \in \{1, \ldots, I\}$	Function approximation point indexes

Constants:

$W \in \mathbb{N}$	Number of wavelengths in any fibre
$V_d \in \mathbb{N}$	Volume of wavelengths to be realised for demand d
$n_d^{\mathrm{src}} \in \{1, \ldots, N\}$	Index of the source node for demand d
$n_d^{\mathrm{dst}} \in \{1, \ldots, N\}$	Index of the destination node for demand d

$K_i \in \mathbb{R}_+$	Coefficient for approximation function i
$C_i^0 \in \mathbb{R}_+$	Offset for approximation function i
$C_{\{n,m\}}^{\text{duct}} \in \mathbb{R}_+$	Cost of constructing the duct on link $\{n,m\}$
$C_{\{n,m\}}^{\text{fibre}} \in \mathbb{R}_+$	Cost of deploying a fibre on link $\{n,m\}$

Variables:

$x_{dnm} \in \mathbb{N}_0$	Flow of demand d along edge (nm)
$u_{\{n,m\}} \in \mathbb{N}_0$	Number of required fibre pairs on link $\{n,m\}$
$\delta_{\{n,m\}} \in \{0,1\}$	Number of required ducts on link $\{n,m\}$
$z_n \in \mathbb{R}_+$	Number of required ports in node n
$\gamma_n \in \{0,1\}$	Number of required cores in node n
$v_n \in \mathbb{R}_+$	Cost of ports and switching at node n

Objective:
Minimise

$$\sum_{\{n,m\}} (C_{\{n,m\}}^{\text{duct}} \cdot \delta_{\{n,m\}} + C_{\{n,m\}}^{\text{fibre}} \cdot 2u_{\{n,m\}}) + \sum_n (C^{\text{core}} \cdot \gamma_n + v_n),$$

subject to
Constraints:

$\sum_{m:(nm)\in\mathcal{E}} x_{dnm} - \sum_{m:(mn)\in\mathcal{E}} x_{dmn} = \begin{cases} V_d, \text{ if } n = n_d^{\text{src}} \\ -V_d, \text{ if } n = n_d^{\text{dst}} \\ 0, \text{ otherwise} \end{cases}$	$\forall d, n$	Flow conservation for paths: what goes into node n must come out again
$\sum_d x_{dnm} + x_{dmn} \leq W u_{\{n,m\}}$	$\forall \{n,m\}$	Calculate the required number of fibre pairs on $\{n,m\}$
$\delta_{\{n,m\}} = 0 \Rightarrow u_{\{n,m\}} = 0$	$\forall \{n,m\}$	Without a duct, there can be no fibres on link $\{n,m\}$
$V(n) + \sum_{\{n,m\}} 2u_{\{n,m\}} = z_n$	$\forall n$	The required number of ports at node i is the sum of add / drop fibres and neighbour node fibres

$\gamma_n = 0 \Rightarrow z_n = 0$	$\forall n$	Without a core, there can be no ports at node n
$K_i z_n + C_i^0 \leq v_n$	$\forall i, n$	The port and switching cost of a node n with a core is bounded from below by linear approximation function i

Where the add/drop fibre count at node n is calculated by $V(n) = 2\left\lceil \frac{\sum_{d \in D_n} V_d}{W} \right\rceil$ and $D_n = \left\{ d \in D \mid n = n_d^{\text{src}} \vee n = n_d^{\text{dst}} \right\}$. We note that if a node is a source or destination for at least one demand then $\gamma_n \neq 0$, so for networks with demands from or to all nodes we can entirely eliminate γ_n from the ILP program.

The ILP program in the link-path formulation. We shall now describe the components of the ILP program in the link-path formulation for the two network design variants, ND and SBP+NRPC. They are based on the concept of the set of network *states* $s = 0, \ldots, S$ described in Section 2.1. Again, in the link-path formulation a number of primary paths have been precalculated for each demand, represented by a three-dimensional binary matrix A which indicates which links are used by which paths. For each primary path a number of backup paths—failure disjoint with the primary path—have also been precalculated and represented in a four-dimensional binary matrix B.

Indices:

$d \in \{1, \ldots, D\}$	Traffic demands
$s \in \{0, \ldots, S\}$	Network states
$n \in \{1, \ldots, N\}$	Network nodes
$l \in \{1, \ldots, L\}$	Network links
$p \in \{1, \ldots, P_d\}$	Primary paths realising demand d
$q \in \{1, \ldots, Q_d^p\}$	Backup paths protecting demand d, path p
$i \in \{1, \ldots, I\}$	Function approximation point indexes

Constants:

$W \in \mathbb{N}$	Number of wavelengths in any fibre
$V_d \in \mathbb{N}$	Volume of wavelengths to be realised for demand d

$A_{ldp} \in \{0,1\}$	Primary link-path coefficient: 1 if path p for demand d uses link l
$B_{ldpq} \in \{0,1\}$	Backup link-path coefficient: 1 if path p for demand d can be protected by backup path q that uses link l
$G^s_{dp} \in \{0,1\}$	Path survival coefficient: 1 if path p of demand d works in state s
$J_{ln} \in \{0,1\}$	Node-link incidence coefficient: 1 if node n is incident with link l
$K_i \in \mathbb{R}_+$	Coefficient for approximation function i
$C^0_i \in \mathbb{R}_+$	Offset for approximation function i
$C^{\text{fibre}}_l \in \mathbb{R}_+$	Cost of deploying a fibre on link l
$C^{\text{duct}}_l \in \mathbb{R}_+$	Cost of constructing the duct on link l

Variables:

$x_{dp} \in \mathbb{N}_0$	Primary flow of demand d on path p
$y_{dpq} \in \mathbb{R}_+$	Backup flow on path q protecting primary flow x_{dp}
$\beta_{dpq} \in \{0,1\}$	Backup indicator: 1 if path p supplying demand d is backed up by path q
$u_l \in \mathbb{N}_0$	Number of required fibre pairs on link l
$\delta_l \in \{0,1\}$	Number of required ducts on link l
$z_n \in \mathbb{R}_+$	Number of required ports in node n
$\gamma_n \in \{0,1\}$	Number of required cores in node n
$v_n \in \mathbb{R}_+$	Cost of ports and switching at node n

Objective:
Minimise

$$\sum_l (C^{\text{duct}}_l \cdot \delta_l + C^{\text{fibre}}_l \cdot 2u_l) + \sum_n (C^{\text{core}} \cdot \gamma_n + v_n),$$

subject to

Constraints:

$$\sum_p x_{dp} = V_d \quad \forall d$$

The total volume of demand d must be supplied by various paths

$\sum_q y_{dpq} = x_{dp}$	$\forall d, p$	Satisfy backup demands
$\sum_q \beta_{dpq} = 1$	$\forall d, p$	Each path must have exactly one backup path
$\beta_{dpq} = 0 \Rightarrow y_{dpq} = 0$	$\forall d, p, q$	Ensure that x_{dp} uses only one backup path
$\sum_d \sum_p \left(\begin{matrix} A_{ldp} x_{dp} + \\ (1 - G^s_{dp}) \sum_q B_{ldpq} y_{dpq} \end{matrix} \right) \leq W u_l$	$\forall l, s$	Calculate the required number of fibre pairs on link l in state s
$\delta_l = 0 \Rightarrow u_l = 0$	$\forall l$	Without a duct, there can be no fibres on link l
$V(n) + \sum_l 2 J_{ln} u_l = z_n$	$\forall n$	The required number of ports at node i is the sum of add/drop fibres and neighbour node fibres
$\gamma_n = 0 \Rightarrow z_n = 0$	$\forall n$	Without a core, there can be no ports at node n
$K_i z_n + C^0_i \leq v_n$	$\forall i, n$	The cost of node n is bounded from below by linear approximation function i

Where again $V(n) = 2 \left\lceil \frac{\sum_{d \in D_n} V_d}{W} \right\rceil$ and $D_n = \left\{ d \in D \mid n = n^{\text{src}}_d \vee n = n^{\text{dst}}_d \right\}$.

Integrated node and link optimisation by SAL

The only changes needed to the SAL heuristic described in Section 2.5 is to include the node cost when computing the cost of colour tokens in the solution (cf. Figure 2.11 on page 50). This is easily computed based on the sum of fibres in the links incident to the node.

3.3.4 Evaluating optimisation method quality

In this section we shall report the results of some experiments with running the ILP programs and the heuristic on various network and traffic examples. The ILP solver used here is the GAMS & CPLEX suite mentioned in Section 2.4.

Networks and traffic demands

We use the following randomly generated fully connected networks of various sizes:

Network	1	2	3	4	5	6
Links	10	45	105	190	300	435
Nodes	5	10	15	20	25	30

The duct construction costs are proportional to the Euclidean distance, and the fibre deployment costs are $1/20$th of the duct cost. Traffic is uniform, that is, the same demand volume for all node pairs, and all experiments operate with $W = 8$ wavelengths per fibre.

Estimating greenfield optimisation tool quality

We now perform some experiments to get an estimate of the quality of the tools for optimising greenfield network design.

Given the network and traffic demand examples described in the preceding section, we fix the node cost parameters $C^{\text{switching}} = 1$, $C^{\text{port}} = 10$, $C^{\text{core}} = 100$, and then determine what the total network cost is for the two network design tasks, ND and SBP+NRPC, and three different network design tools: The ILP link-path program, the ILP arc-flow program and the SAL heuristic.

Each experiment is run with the SAL optimiser 3 times for 20 minutes, using $(k_s, k_d, k_b) = (5, 2, 0)$ and $(8, 2, 3)$ for the ND and SBP+NRPC tasks, respectively, whereupon the average total network cost is calculated. The ILP link-path programs are run for up to 20 minutes, and the ILP arc-flow program for 1 hour, on a 440MHz HP Series 9000 Model J7000 with 1Gb RAM per optimisation process.

The results are shown in Figures 3.26 and 3.27; 95% confidence intervals are indicated by two horizontal lines.

The ILP solver times out on the ND arc-flow program for the 45 link network with the best found solution value shown in the figure, while it does not reach any feasible solutions for larger networks. For the SBP+NRPC link-path program, the ILP solver times out for networks with 45 or more links, returning no solution. The SAL optimiser reaches ND solutions for the 435 link network that cost 14% more than the value found using the ILP link-path program.

As the ILP arc-flow program timed out already at 45 links we cannot accurately determine the quality of the paths calculated by the PPG. Similarly, as the ILP link-path program only returns results for the 10 link network, we cannot determine the relative quality of the SAL heuristic for the SBP+NRPC design task.

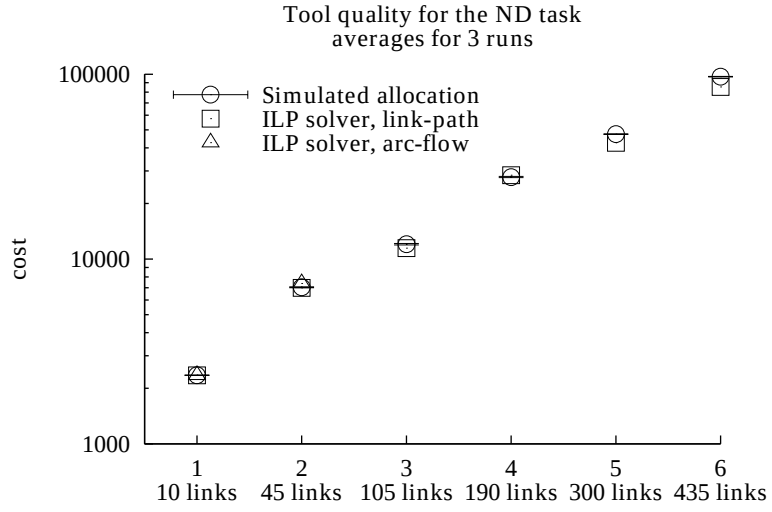


Figure 3.26: Optimisation tool quality for the ND design task

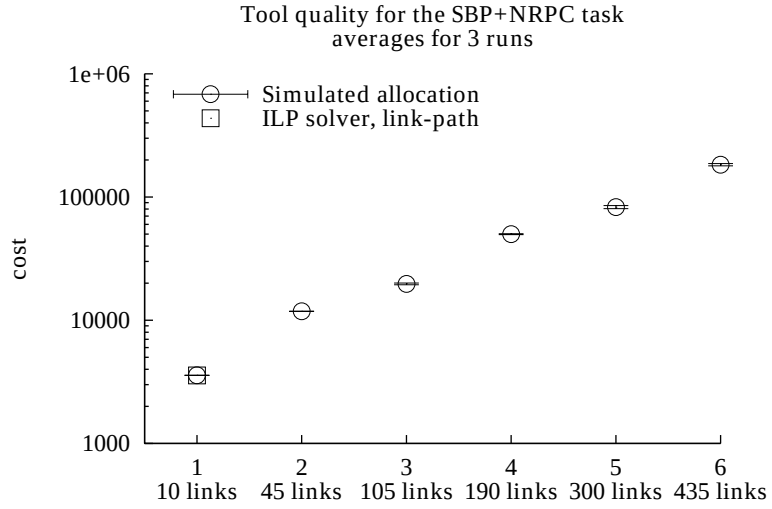


Figure 3.27: Optimisation tool quality for the SBP+NRPC design task

Staged versus integrated optimisation

Next, we compare staged optimisation, where node costs are calculated in a post-processing stage after link-based optimisation, with integrated node and link optimisation.

Fixing $C^{\text{port}} = 10$ and $C^{\text{core}} = 100$ we consider just the 300 link network for the two network design tasks, ND and SBP+NRPC, and vary $C^{\text{switching}}$ between 0 and 1. For each value of $C^{\text{switching}}$ we run the heuristic 3 times with the staged optimisation and 3 times with the integrated optimisation, determining $\frac{\text{cost}^{\text{integrated}}}{\text{cost}^{\text{staged}}}$ (i.e., how much cheaper integrated optimisation is with respect to sequential optimisation) and the relative cost of nodes, fibres and ducts.

Using the same computer as in the preceding section, we obtain the results shown in Figures 3.28 and 3.29. We see that integrated optimisation is up to

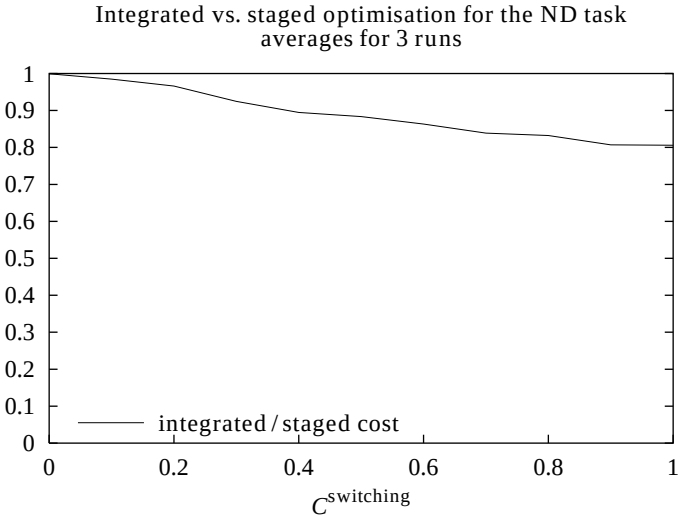


Figure 3.28: Staged versus integrated optimisation for the ND task on the 300 link network

20% cheaper on overall network cost than staged optimisation for the ND task. However, for the SBP+NRPC task there is little difference, at most 12%, and for small values of $C^{\text{switching}}$ the network cost can even increase when performing integrated optimisation.

Figures 3.30 and 3.31 show the node, fibre and duct costs as fractions of the total network cost. We can clearly see that integrated optimisation attempts to

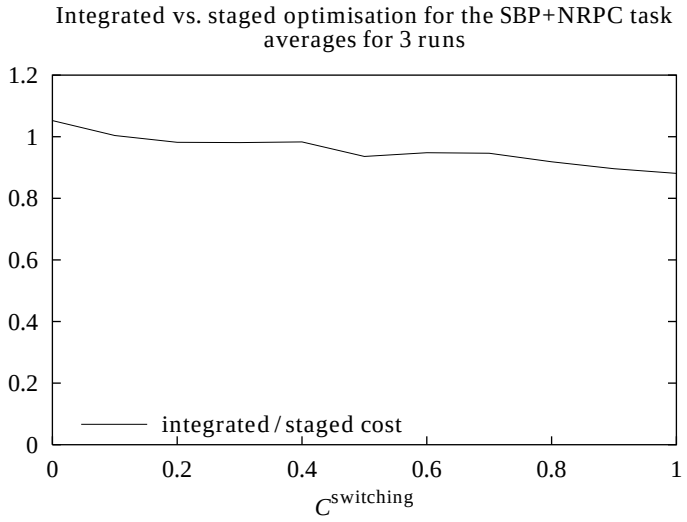


Figure 3.29: Staged versus integrated optimisation for the SBP+NRPC task on the 300 link network

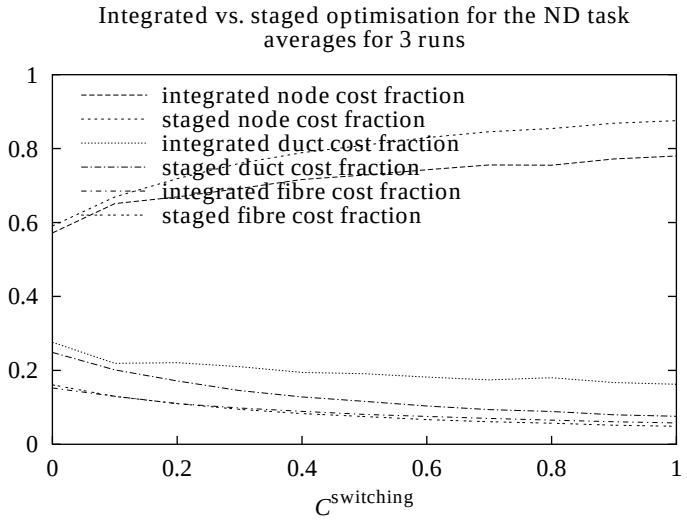


Figure 3.30: Node, fibre and duct cost fractions of the total network cost for the ND task with the 300 link network

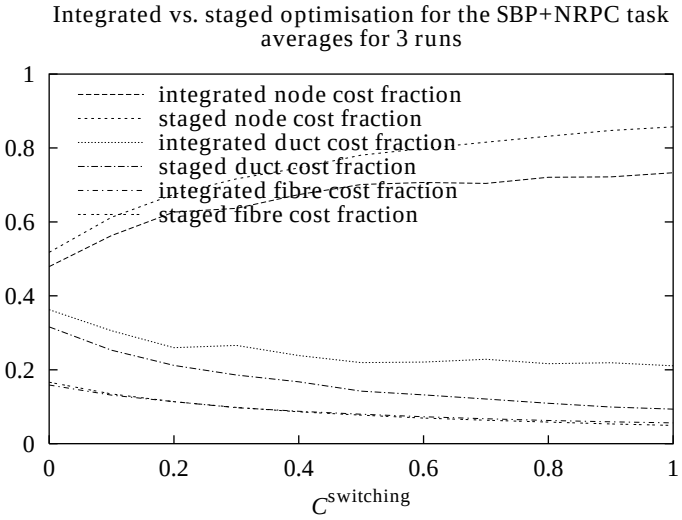


Figure 3.31: Node, fibre and duct cost fractions of the total network cost for the SBP+NRPC task with the 300 link network

reduce the node costs, sacrificing cheap duct costs. This effect is confirmed in Figure 3.32 on the following page which shows an example of the resulting topology from staged and integrated optimisation. We also note that the resulting network topologies are very similar for staged and integrated optimisation; the latter reduces the largest nodes slightly, adding a handful of lightly loaded direct links.

Optimising node switch deployment

In the final experiments we attempt to determine how good the tools are at handling a *non*-greenfield network design task: Given the nodes and the actual links (i.e., not all the potential links) of a network, we consider the task of routing the traffic demands to minimise node equipment costs. Fixing $C^{\text{port}} = 10$, $C^{\text{core}} = 100$, $C_l^{\text{duct}} = 0$, and letting C_l^{fibre} be proportional to the Euclidean distance, we consider the Pan-European optical network shown in Figure 3.19 on page 102 for the gravitational traffic demands shown in Table 3.5 on page 127. Note that we model symmetric connections, so only demands from the upper triangular matrix are used.

We vary $C^{\text{switching}}$ between 0 and 1 and use the SAL heuristic to perform the

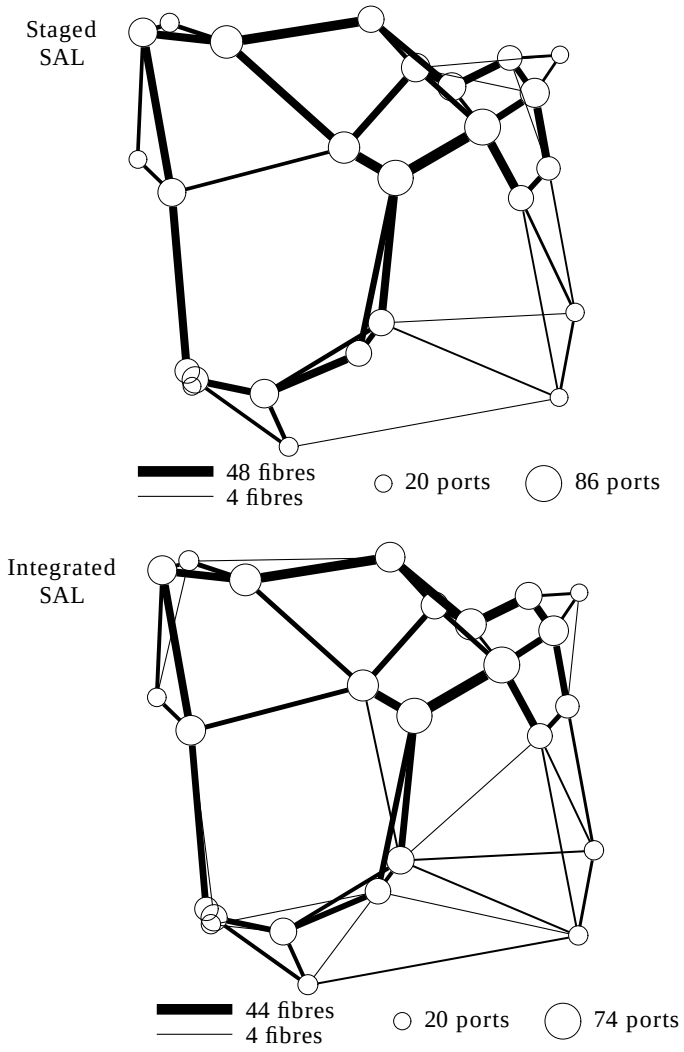


Figure 3.32: Results for the 300 link network performing the SBP+NRPC task with $C^{\text{switching}} = 0.3$, $C^{\text{port}} = 10$ and $C^{\text{core}} = 100$

	Helsinki	Stockholm	Oslo	Copenhagen	Berlin	Amsterdam	Dublin	Prague	Luxemburg	Brussels	London	Vienna	Zurich	Paris	Zagreb	Athens	Rome	Madrid	Lisbon
Helsinki	0	1	1	1	3	1	1	1	1	1	2	1	1	2	1	1	2	2	1
Stockholm	1	0	1	1	4	1	1	1	1	1	3	1	1	3	1	1	3	2	1
Oslo	1	1	0	1	2	1	1	1	1	1	2	1	1	2	1	1	2	1	1
Copenhagen	1	1	1	0	3	1	1	1	1	1	2	1	1	2	1	1	2	2	1
Berlin	3	4	2	3	0	7	2	5	1	5	26	4	4	25	2	5	24	17	5
Amsterdam	1	1	1	1	7	0	1	1	1	1	5	1	1	5	1	1	5	4	1
Dublin	1	1	1	1	2	1	0	1	1	1	2	1	1	2	1	1	2	1	1
Prague	1	1	1	1	5	1	1	0	1	1	4	1	1	4	1	1	3	3	1
Luxemburg	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
Brussels	1	1	1	1	5	1	1	1	1	0	4	1	1	4	1	1	3	3	1
London	2	3	2	2	26	5	2	4	1	4	0	3	3	18	2	4	18	12	4
Vienna	1	1	1	1	4	1	1	1	1	1	3	0	1	3	1	1	3	2	1
Zurich	1	1	1	1	4	1	1	1	1	1	3	1	0	3	1	1	3	2	1
Paris	2	3	2	2	25	5	2	4	1	4	18	3	3	0	2	4	18	12	4
Zagreb	1	1	1	1	2	1	1	1	1	1	2	1	1	2	0	1	2	1	1
Athens	1	1	1	1	5	1	1	1	1	1	4	1	1	4	1	0	4	3	1
Rome	2	3	2	2	24	5	2	3	1	3	18	3	3	18	2	4	0	12	3
Madrid	2	2	1	2	17	4	1	3	1	3	12	2	2	12	1	3	12	0	2
Lisbon	1	1	1	1	5	1	1	1	1	1	4	1	1	4	1	1	3	2	0

Table 3.5: Traffic demand matrix for the Pan-European optical network in units of wavelengths

two network design tasks, ND and SBP+NRPC. Each experiment is run 3 times, using the same computers as in Section 3.3.4, and the average total network cost is calculated.

The results are shown in Figures 3.33 and 3.34, with node and fibre cost fractions shown in Figures 3.35 and 3.36. The same general picture as in the preceding section is observed: For the ND design task up to 13% of the total network cost can be saved by using integrated optimisation, while there are no significant savings for the SBP+NRPC design task. Example optimised network topologies are shown in Figure 3.37 on page 131; again with only minor variations between the two methods.

3.3.5 Conclusions

In this work we presented a general node cost model including a quadratic term in the number of node ports and extended the PPG and SAL heuristic developed in Sections 2.3 and 2.5.

Experiments showed that for the ND design task the extended heuristic was able to design example networks with a cost at most 14% more than when using the CPLEX ILP optimiser. Generally, integrated optimisation reduces node costs, which can lead to overall greenfield network deployment cost savings of up to 20%, depending on the scaling factor of the quadratic term.

The results for the SBP+NRPC design task are not decisive, but the present SAL heuristic does not seem able to achieve significant savings for this design task.

Further research should investigate whether improved design methods could reduce the network cost for the SBP+NRPC design task. The similarity of the resulting network topologies in Figures 3.32 and 3.37 could suggest that the paths generated by the PPG are very similar and restrict the opportunities the SAL heuristic is left for reducing network costs.

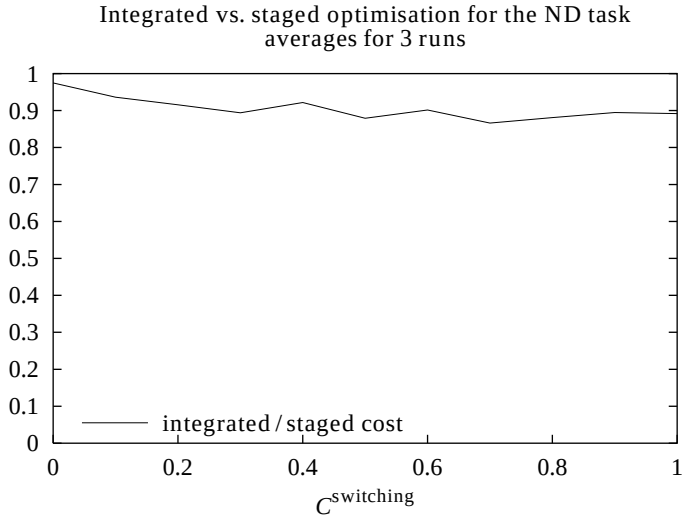


Figure 3.33: Relative cost savings using the optimisation tool to route static traffic demands in the Pan-European optical network with $C^{\text{port}} = 10$, $C^{\text{core}} = 100$

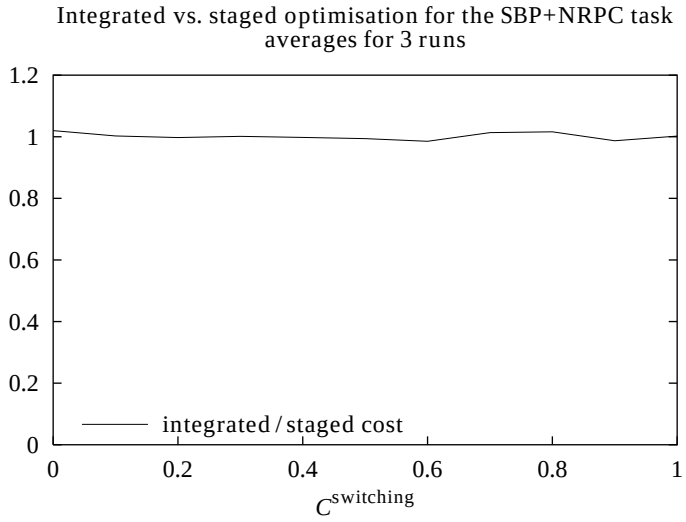


Figure 3.34: Relative cost savings using the optimisation tool to route static traffic demands in the Pan-European optical network with $C^{\text{port}} = 10$, $C^{\text{core}} = 100$

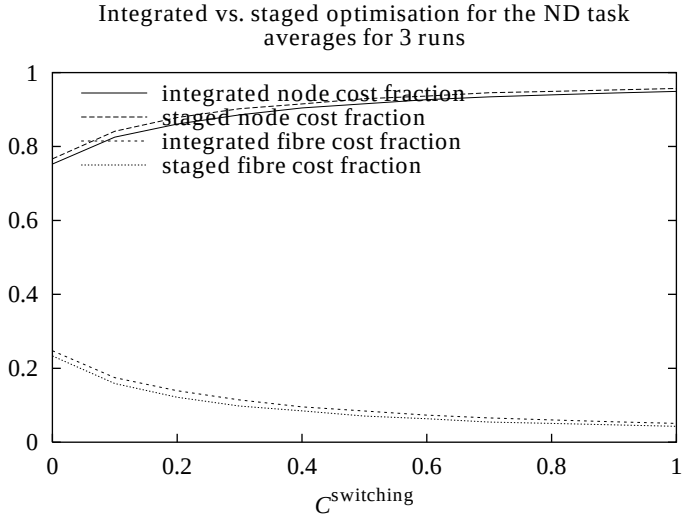


Figure 3.35: Cost fractions of the total network cost for the Pan-European network

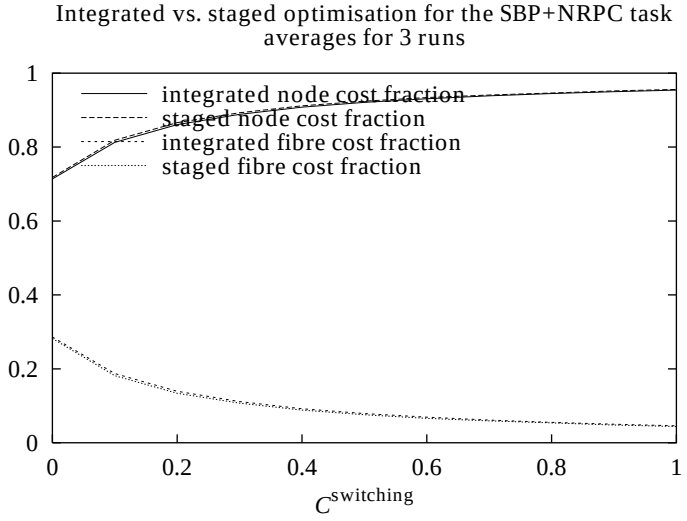


Figure 3.36: Cost fractions of the total network cost for the Pan-European network

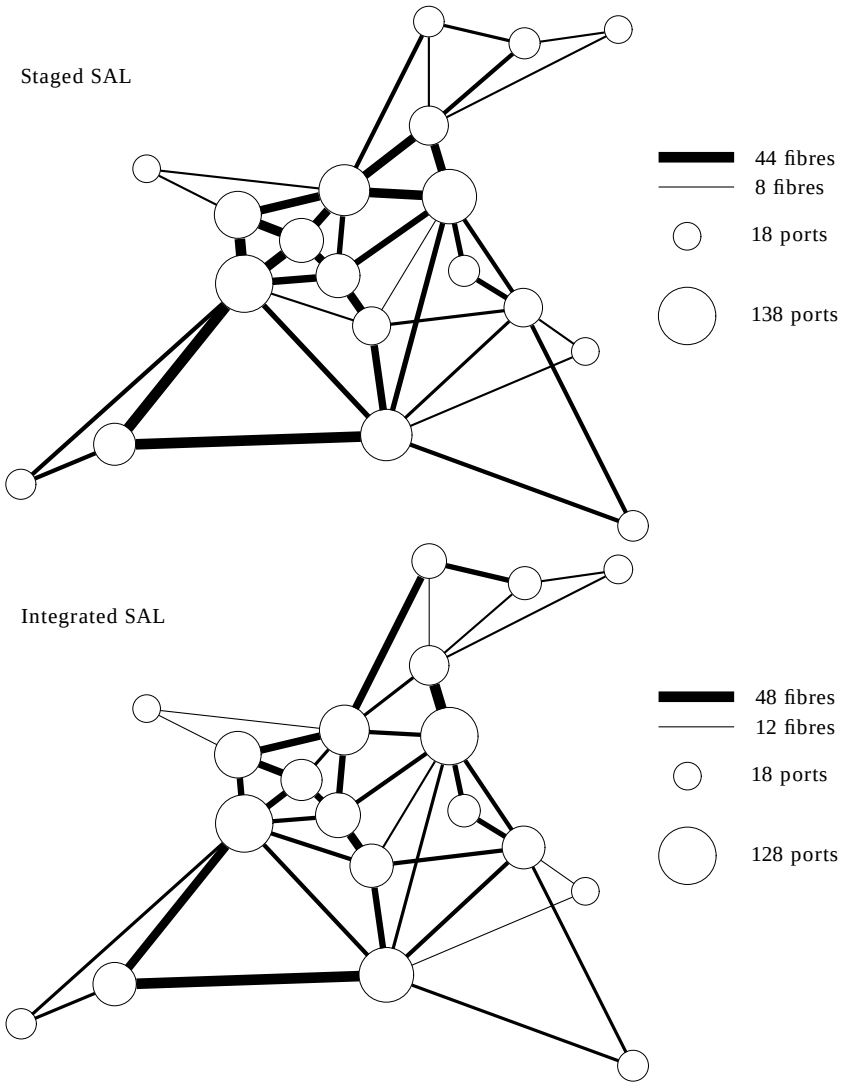


Figure 3.37: Examples of resulting networks using the integrated optimisation tool to route static traffic demands for the SBP+NRPC design task in the Pan-European optical network where $C^{\text{switching}} = 0.5$, $C^{\text{ports}} = 10$ and $C^{\text{core}} = 100$

3.4 Optimising MS-SPRing networks

A tremendous growth of the global communication networks has taken place in the last decade, both in terms of investment and capacity. This growth has made network planning a vital activity for the telecommunication companies. An important part of the network planning is assuring protection of the networks, so that the communication flow can recover quickly from defined error scenarios. Protection against error scenarios is important because companies and people increasingly depend on the network, and failures are thus more costly. Further, a single cable cut may affect many people because of the large capacity of each fibre connection; hence reliability has become a competitive parameter for the telecommunication companies.

Even though the technological development has been fast in recent years, standardisation has progressed at a much slower rate, making protection hard and expensive. The main protection methods today are $1+1$ protection and ring protection (Sexton and Reid, 1997), and while both these mechanisms are fairly expensive in the required hardware costs, they have the common advantage over general $1:n$ path protected mesh networks of having a very fast protection switching time. Furthermore, the management of mesh network path protection has not yet been standardised—consequently ring protection has received most attention, being also the cheapest approach given its simplicity and sharing of protection capacity.

A number of articles have been written about the construction of ring networks (Gardner et al., 1995). In the work by Cosares et al. (1995) the implemented model has been decomposed so that the solution cannot be claimed to be optimal. Morley and Grover (1999) give an optimal model of a ring network, but only a fraction of it is solved, and hence the optimum remains unknown. Furthermore, the cost metric only concerns the link costs. In the work of Gardner et al. (1994) a heuristic for constructing a ring cover for a mesh network is given, and good results are reported; Luss et al. (1998) present another scheme for the same task. All this work, except that of Morley and Grover, and Wuttisittikulkij et al. (2000), assumes that the routing (i.e., the traffic on each link) is given and the ring network has to be constructed. Morley and Grover consider the routing of the traffic and the ring construction at the same time—however, they also ignore the costs of the node equipment. Wuttisittikulkij et al. construct several different heuristics for capacity design of mesh networks and of ring networks, given a mesh network. These heuristics do not take node costs into account, nor do they integrate the selection of paths and rings, but are promising algorithms for optimisation of larger ring networks.

In this work we focus on two aspects:

Quantitative comparison of *real-world* data: We consider networks actually in operation, and perform experiments using real prices for switch node components and fibre links. Key questions include

- What are the differences between networks designed by human network planners and optimal networks designed by optimisation programs?
- How is the cost of the network distributed between link costs and node costs?
- How should the components be priced, for protected mesh networks to be competitive with ring networks?

Comparison of IR and PR: We consider two methods for automatic ring design: integrated routing (IR), where path and ring selection are integrated into one optimisation, and prerouting (PR) where paths are routed prior to ring selection. Key questions include

- Quantitative: What are the cost differences?
- Qualitative: What do the solutions look like?

If we want to investigate these questions, we need a more detailed model of the networks than the models presented in the previously mentioned articles, and we need realistic examples to perform experiments.

In this work we will formulate linear programs which model ring networks and path protected mesh networks; both models take routing as well as protection into account, and furthermore use realistic costs regarding both links and nodes. The ring network linear programming (LP) model is based on the classic LP flow model described in Section 2.4.1 on page 34, extended to include rings, and the mesh network model is a standard link-path LP model, cf. Sections 2.4.1 and 2.4.2.

Since the problems are very hard and the models quite detailed, we should not expect the LP models to be solvable except for small networks. Both problems are inherently ILP, but we can only solve their relaxation—hence a simple rounding scheme is used to obtain integer solutions.

The results of this work suggest that the price of mesh network components must be reduced significantly to be competitive with ring based networks, and also that manual network design does not necessarily lead to the most cost-efficient designs.

3.4.1 Models

We consider a design problem where the traffic demands are static and given in a matrix as a number of VC-4 connections (end-to-end bandwidths), for each node

pair. The topology is given as a set of nodes and links, but the problem is *uncapacitated* in the sense that each link can be assigned as many fibres as necessary, and each node as many switch components as necessary. The cost of a link is proportional to the capacity used, so we do not impose an initial cost whenever a new fibre is needed. Furthermore, contrary to the greenfield model used in Chapter 2, there is no duct cost involved. The objective is to design a routing and a network with node and link capacities that support the routed connections and provide 100% protection against single link failures.

The ring network technology considered in this work is the MS-SPRing system described in Section 1.4.4 on page 15. In order to compare MS-SPRing protection with mesh network protection switching, we formulate several linear programs: First, we formulate two ring models: one where the routing of traffic demands is integrated with the ring selection, and one where the routing is performed prior to ring selection. Then, for comparison, we describe a mesh network model for routing paths and allocating protection capacity.

Ring model—integrated routing

In this section we present a linear program which models a ring network. The characteristic feature of this model is that the optimisation of the paths is performed *concurrently* with the selection of which rings to use. Note that we will not model protected connections *between* the rings—e.g., Drop and Continue (Sexton and Reid, 1997)—in the present work. For this reason, the ring network solution is only protected against link failures and not against ring-connecting node failures.

The overall structure of the model is that of a standard network flow model (Cook et al., 1998), where the constraints have been refined by parametrising them over a set of potential rings. Constraints are added for modeling the ring architecture, where all links in a ring must have the same capacity, and also constraints that relate the paths with the amount of traffic that is switched between rings. The total cost of the network is then calculated on the basis of the required link and switch capacity.

There are two specific components in the MS-SPRing model: nodes and rings:

Nodes Each node consists of three parts, cf. Figure 3.38 on the facing page. A number of cores (frame and switching fabric), the input/output equipment (tributary cards), and the transit equipment (line cards). Each line card is connected to a fibre. For each node we define three quantities: The *transit capacity*, z^{T}, the *I/O flow*, y, and the *number of cores*, z^{C}. The transit capacity is the total number of fibres on the rings of which the node is a part. Thus twice as many line cards are necessary because each node is connected to two other nodes in each ring. The I/O flow is the amount of traffic that

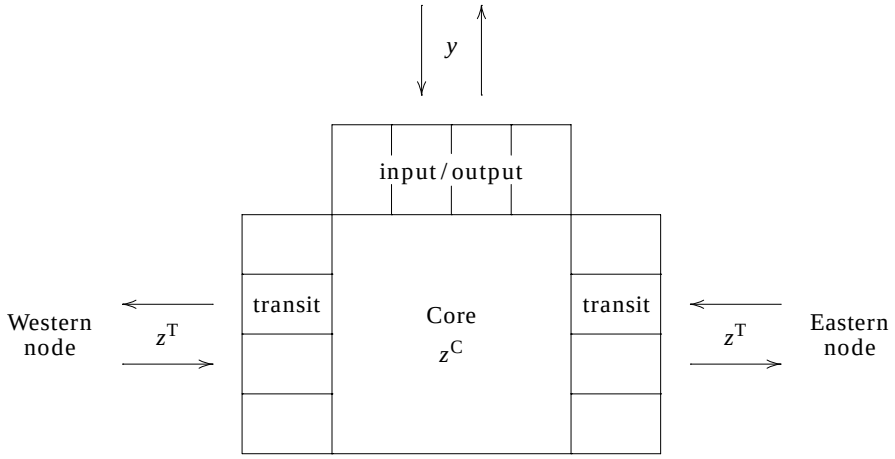


Figure 3.38: MS-SPRing node model

switches from one ring to another, plus the amount of traffic that is started or terminated at the node.

The cost of a node is then the sum of the costs of the three different parts. For the experiments we use real-world costs, but due to confidentiality concerns, we only give the relative costs of the various node components here (note that one STM-1 link unit can support one VC-4 connection):

MS-SPRing component	*Max # per core*	*Fractional cost for one component*
Core	1	$C^{\text{core}} = 1.00$
Line card (4 $\times$ STM-1)	8	$C^{\text{linecard}} = 0.36$
Tributary card (4 $\times$ STM-1)	4	$C^{\text{tribcard}} = 0.38$

Rings The MS-SPRing architecture comprises fully bidirectional rings (Sexton and Reid, 1997). In an MS-SPRing system, each link of a ring has the same number of fibres for transporting the nominal, bidirectional flow. An identical number of fibres are allocated as backup on each link, and when a link fails, the nodes at each end of the failure switch over to use all the backup links around the ring in the opposite direction. All the connections carried by the ring share the backup capacity, and protection switching is done locally, making it very fast. We model the capacity of each ring as twice the maximal flow on any of the links of the ring.

Figure 1.7 on page 18 gives an example of how the protection mechanism works. In the ring, north-south and east-west-going signals are interrupted

by a north-east link break. The ring then automatically reroutes the signals the opposite way round the ring on the backup fibres. The paths that are affected by the failure are not entirely rerouted from end to end, but only around the broken link, so this is an example of *link protection*.

Based on the prices for the node components we calculate:

- Transit cost per VC-4: $C^{\text{transit}} = \frac{2}{4} \cdot C^{\text{linecard}}$: One transit connection (either a normal or protection) takes up 1 of 4 STM-1 units connected to the preceding node in the ring and 1 of 4 STM-1 units connected to the next node.
- I/O costs per VC-4: $C^{\text{IO}} = \frac{1}{4} \cdot C^{\text{tribcard}}$: When entering or exiting a node, 1 of 4 ports on the tributary card is used.

These costs are used in the linear program which is given in detail in the following. The model uses a precalculated set of R potential rings, represented by three constant vectors F, G, H, defined in the following text. First we describe the indices, constants and variables, and then we present the objective and constraints.

Indices:

$d \in \{1, \ldots, D\}$	Demands between pairs of nodes
$(nm) \in \mathcal{E} \subseteq \{1, \ldots, N\}^2$	Directed edges
$\{n, m\} \in \mathcal{L} \subseteq \mathcal{P}(\{1, \ldots, N\})$	Undirected links
$r \in \{1, \ldots, R\}$	Potential rings which may be used.

Constants:

$V_d \in \mathbb{N}$	Volume of VC-4 connections to be realised for demand d
$n_d^{\text{src}} \in \{1, \ldots, N\}$	Starting node of demand d
$n_d^{\text{dst}} \in \{1, \ldots, N\}$	Terminating node of demand d
$C_{\{n,m\}}^{\text{link}} \in \mathbb{R}_+$	Transmission cost for each STM-1 transported on link $\{n, m\}$
$C^{\text{transit}} \in \mathbb{R}_+$	Transit cost for each VC-4 passing through a node
$C^{\text{IO}} \in \mathbb{R}_+$	I/O (ring shift) costs for each VC-4, for all nodes
$C^{\text{core}} \in \mathbb{R}_+$	Cost of a core unit
$F_{(nm)}^r \in \{0, 1\}$	Set to 1 if ring r uses directed edge (nm), otherwise 0
$G_{\{n,m\}}^r \in \{0, 1\}$	Set to 1 if ring r uses undirected link $\{n, m\}$, otherwise 0
$H_n^r \in \{0, 1\}$	Set to 1 if ring r uses node n, otherwise 0

Variables:

$x^r_{dnm} \in \mathbb{R}_+$	Flow for demand d on the directed edge (nm) in ring r
$y_n \in \mathbb{R}_+$	I/O flow at node n
$y^{dr}_{n+} \in \mathbb{R}_+$	Flow for demand d on ring r into node n
$y^{dr}_{n-} \in \mathbb{R}_+$	Flow for demand d on ring r out of node n
$z^{\mathrm{R}}_r \in \mathbb{R}_+$	Required fibre capacity for ring r including protection capacity
$z^{\mathrm{T}}_n \in \mathbb{R}_+$	Required transit capacity (including protection) at node n. The required number of ports is twice this number, one for each direction in the ring
$z^{\mathrm{C}}_n \in \mathbb{R}_+$	Required number of cores in node n
$u_{\{n,m\}} \in \mathbb{R}_+$	Total required capacity on link $\{n, m\}$

All flows are measured in VC-4 units and all capacities except z^{C}_n are measured in STM-1 units. Note that all variables are non-negative reals, as opposed to integers, which enables the linear program solver to find an optimal solution within reasonable time.

Linear program Based on the preceding definitions we now specify the linear program for minimising the costs of installing a ring network in an existing topology. The total cost is a sum of costs for links, cores, line cards and tributary cards:

Objective: minimise

$$cost_{\mathrm{ring}} = \sum_{\{n,m\}} C^{\mathrm{link}}_{\{n,m\}} \cdot u_{\{n,m\}} + \sum_n C^{\mathrm{core}} \cdot z^{\mathrm{C}}_n + \sum_n C^{\mathrm{transit}} \cdot z^{\mathrm{T}}_n + \sum_n C^{\mathrm{IO}} \cdot y_n$$

Constraints:

$$\begin{array}{r} \Sigma_r \Sigma_{m:(nm)\in\mathcal{E}} F^r_{(nm)} \cdot x^r_{dnm} \\ -\Sigma_r \Sigma_{m:(mn)\in\mathcal{E}} F^r_{(mn)} \cdot x^r_{dmn} \end{array} = \begin{cases} V_d & \text{if } n = n^{\mathrm{src}}_d \\ -V_d & \text{if } n = n^{\mathrm{dst}}_d \\ 0 & \text{otherwise} \end{cases} \quad \forall n, d$$

Flow conservation: what goes into node n must come out again

$$2 \cdot \sum_d G^r_{\{n,m\}} \cdot (x^r_{dnm} + x^r_{dmn}) \leq z^{\mathrm{R}}_r \qquad \forall r, \{n, m\}$$

Ring capacity is twice (for protection) the maximal link flow

$$\sum_r G^r_{\{n,m\}} \cdot z^{\mathrm{R}}_r = u_{\{n,m\}} \qquad \forall \{n,m\}$$ Link capacity is the sum of capacities of all rings using it

$$\sum_r H^r_n \cdot z^{\mathrm{R}}_r = z^{\mathrm{T}}_n \qquad \forall n$$ Transit capacity for node n is the sum of all its rings' capacities

$$(\star) \quad \begin{array}{r} \sum_{m:(nm)\in\mathcal{E}} F^r_{(nm)} \cdot x^r_{dnm} \\ -\sum_{m:(mn)\in\mathcal{E}} F^r_{(mn)} \cdot x^r_{dmn} \end{array} = y^{dr}_{n+} - y^{dr}_{n-} \quad \forall n,r,d$$ Calculate input or output flow

$$\sum_d \sum_r \left(y^{dr}_{n+} + y^{dr}_{n-} \right) = y_n \qquad \forall n$$ Calculate the I/O flow y_n

$$z^{\mathrm{C}}_n \geq \frac{z^{\mathrm{T}}_n}{16} \wedge z^{\mathrm{C}}_n \geq \frac{y_n}{16} \qquad \forall n$$ Ensure enough switch cores for transit and I/O ports

Note that for given n, r we have $F^r_{(nm)} \neq 0$ for at most one j (and similarly for $F^r_{(mn)}$), because there is exactly one ingoing and one outgoing link for each node in ring r. Thus, all the left hand sides of constraints $(\star)$ consist of no terms if node n is not in ring r, and exactly two terms if node n *is* in ring r: $x^r_{dnm} - x^r_{dmn}$. This is exactly the volume of flow for demand d entering ring r at node n. If this volume is positive, $y^{dr}_{n+} > 0$, and if it is negative, $y^{dr}_{n-} > 0$. Either way, the I/O flow y_n is calculated by adding the input and output flows.

Ring model—prerouting

The model in the preceding section optimises the routing and the ring selection concurrently. A more popular way of designing ring networks is to split the optimisation into two stages: First, all the demands are routed and then, based on these paths, the rings are selected and their capacities determined (Cosares et al., 1995; Wuttisittikulkij et al., 2000).

It is straightforward to extend the linear program to handle prerouted traffic: We let the route for each demand be given in a two-dimensional binary matrix B such that $B^d_{(nm)} = 1$ indicates that the route for demand d uses link (nm). Then we replace the first constraint (the flow conservation constraint) with the constraint

$$\sum_r x^r_{dnm} = V_d \cdot B^d_{(nm)} \qquad \forall (nm), d.$$

This constraint ensures that if traffic is flowing through a link, given by $B^d_{(nm)}$, it is assigned to one or more rings, given by x^r_{dnm}.

Mesh model

Aiming to compare the cost of the ring based solution, we formulate a linear program which finds an optimal mesh based solution. In this mesh network model, each node is a general DXC, and traffic is protected from single-link failures by path protection: when a link fails, broken paths are re-routed end-to-end. The remaining working link capacity on the broken paths is released and can be used for other connections in the re-routing.

The linear program we use is a standard link-path model which requires pre-calculated paths for each demand as described in Section 2.4.2. One could also construct a linear program using an arc-flow model as described in Section 2.4.1, which does not require pre-calculation of all the paths. We would then relax the x^c_{dnm}, y^c_{dnm}, v^c_d and w^c_d of that section to be nonnegative integers (or reals), which would allow a primary path to be protected by several different backup paths. However, we expect the arc-flow formulation to be too slow in practice.

The link model is the same as for the ring architecture, but the nodes are slightly different because they are modelled as DXCs, cf. Figure 3.39. The DXC

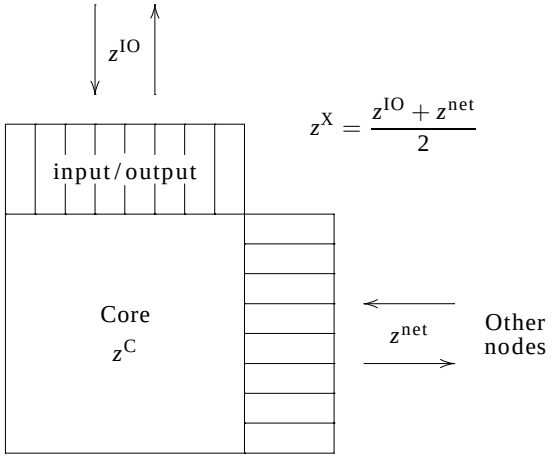


Figure 3.39: DXC node model

looks somewhat similar to the add-drop node in Figure 3.38 on page 135, but the difference is that we do not distinguish between transit traffic and I/O traffic, as

they are both handled by the same port type. If there is so much traffic that more than one DXC core is necessary, we assume that the connections can be made in such a way that each path that passes through the node only uses one of the cores so there is no traffic between them.

Again we do not present absolute costs, but give the costs of the various DXC components relative to the cost of one MS-SPRing core:

DXC component	*Max # per core*	*Fractional cost of one*
Core	1	$C^{\text{core}} = 9.24$
STM-1 magazine (capacity of 8 cards)	16	$C^{\text{magazine}} = 1.10$
Electrical STM-1 card	128	$C^{\text{card}} = 0.14$

As can be seen, networks based on DXC nodes can be significantly more expensive, as each component is more expensive than its MS-SPRing counterpart.

Based on these prices we calculate the switching costs per VC-4: $C^{\text{switching}} = 2 \cdot (\frac{1}{8} \cdot C^{\text{magazine}} + C^{\text{card}})$. Again we use a capacity of two STM-1 per VC-4 signal passing through the node, but this time we do not discriminate between signals which pass through and signals which enter or exit at the node. This price is used in the linear program given in the following sections.

Indices:

$d \in \{1, \ldots, D\}$	Demands between pairs of nodes
$p \in \{1, \ldots, P_d\}$	Paths for demand d
$s \in \{0, \ldots, S\}$	Network states (0 = no failures)
$n \in \{1, \ldots, N\}$	Nodes
$l \in \{1, \ldots, L\}$	Undirected links

Constants:

$V_d \in \mathbb{N}$	Volume of VC-4 connections to be realised for demand d
$C_l^{\text{link}} \in \mathbb{R}_+$	Transmission cost for each VC-4 on link l
$C^{\text{switching}} \in \mathbb{R}_+$	Switching cost for one VC-4 in a node
$C^{\text{core}} \in \mathbb{R}_+$	Core cost per VC-4 connection
$A_{ldp} \in \{0, 1\}$	Set to 1 if path p of demand d uses link l, otherwise 0
$G_{dp}^s \in \{0, 1\}$	Set to 0 if path p of demand d is broken in state s, otherwise 1
$H_{idp} \in \{0, 1\}$	Set to 1 if path p of demand d uses node n, otherwise 0

Variables:

$x_{dp} \in \mathbb{R}_+$	Primary flow on path p of demand d when no links are broken
$y^s_{dp} \in \mathbb{R}_+$	Backup flow added on path p of demand d in state s
$z^{\mathrm{X}}_n \in \mathbb{R}_+$	Required switching capacity (number of switched connections) at node n for all states
$z^{\mathrm{C}}_n \in \mathbb{R}_+$	Required number of cores in node n
$u_l \in \mathbb{R}_+$	Required link capacity for link l in all states

All flows are measured in VC-4 units and all capacities except z^{C}_n are measured in STM-1 units. Again, all variables are non-negative reals, as opposed to integers, to be able to find an optimal solution within reasonable time.
Objective: minimise

$$cost_{\mathrm{mesh}} = \sum_n C^{\mathrm{switching}} \cdot z^{\mathrm{X}}_n + \sum_n C^{\mathrm{core}} \cdot z^{\mathrm{C}}_n + \sum_l C^{\mathrm{link}}_l \cdot u_l$$

Constraints:

$\sum_p x_{dp} = V_d$	$\forall d$	Supply demand d with primary flows
$\sum_p G^s_{dp} \cdot \left(x_{dp} + y^s_{dp}\right) = V_d$	$\forall d, s$	Supply demand d with backup flow in state s
$\sum_d \sum_p H_{idp} \cdot G^s_{dp} \cdot \left(x_{dp} + y^s_{dp}\right) \leq z^{\mathrm{X}}_n$	$\forall n, s$	Ensure sufficient node switching capacity at node n in state s
$z^{\mathrm{C}}_n \geq \frac{z^{\mathrm{X}}_n}{64}$	$\forall n$	Calculate the number of required cores
$\sum_d \sum_p A_{ldp} \cdot G^s_{dp} \cdot \left(x_{dp} + y^s_{dp}\right) \leq u_l$	$\forall l, s$	Ensure sufficient link capacity on link l

Note that each switched connection uses two ports, which is the reason for dividing the switching capacity z^{X} by 64 to find the number of required DXC cores.

Obtaining integer solutions

In general, the solutions obtained from the linear programs are not integral, that is, solutions with an integral number of switches and fibres. To obtain this, we round up the values found by the linear programs.

For the ring network model, we round up the values of variables x^r_{dnm}, and then re-calculate u_l, z^{R}_r, z^{T}_n and y_n, rounding the latter two up to a multiple of 4

because each port handles 4 STM-1 links. Then we calculate the integer number of cores needed for each node: Each core may support only one ring and each ring can have at most 16 VC-4 connections, hence for each ring r in which the node participates, the traffic z_r^{R} is divided by 16 and rounded up. The number of cores needed is then the total sum of these core numbers.

For the mesh network, the flows x_{dp} and y_{dp}^s are rounded up; the resulting link costs can then be calculated in a straightforward manner. For each node we need to calculate the number of cores, the number of magazines and the number of electrical STM-1 cards. Whereas in the ring architecture each node requires different core components for the different rings in which it is participates, one DXC can handle connections to all the neighbouring nodes. The number of cores needed is thus the traffic through the node divided by 64 and rounded up, and the number of magazines is the traffic divided by 4 and rounded up. Finally, the number of STM-1 cards needed is simply the traffic rounded up.

3.4.2 Experimental design

We now use the linear programs to perform a series of experiments which can give us some answers to the key questions mentioned in the start of Section 3.4. The major characteristics of the solutions that we are interested in are

- the cost of the network and its components
- the number of rings in the ring architecture
- the mean and variance of the number of nodes per ring

Network topology & traffic In the experiments we use a real-world network topology, the backbone network of the Danish telecom operator TDC, using realistic link costs. The network has 12 nodes and 25 links, giving a connectivity of $2\frac{25-12+1}{11\cdot 10} \approx 0,25$, an average node degree of 3.8, and it must satisfy 22 demands with a total demand volume of 1734 VC-4s. For each subset of at least 3 and at most 16 nodes, we find the ring connecting them with the least cost; there are 417 such rings, with an average length of 7.6 non-oriented links, which are modeled by 15.2 oriented links.

Even though this network is fairly small, the LP model for the ring network is fairly large; this can be seen by looking at the number of flow variables $\left|x_{dnm}^r\right| = 22 \cdot 417 \cdot 15.2 \approx 140{,}000$.

When a network specialist was given the task of manually designing a ring network in this topology, he selected 3 rings, which cover the entire network,

using only 14 of the potential 25 links. Using only these 3 rings an optimal routing is found using the ring LP and the costs are compared with the automatic ring design. To make a fair comparison, we also perform experiments using only the same 14 links.

For the mesh network LP model, we precalculate *all* possible paths to make sure that the optimal solution is in fact found.

Design method We perform experiments for the two architectures: ring and mesh network, and for two ways of routing paths: IR or PR.

Network element costs There is a significant cost difference between the required amount of node equipment in the ring network and the mesh network. To study the influence of this price difference we simply reduce the combined node costs by a factor α that is varied, e.g. $\alpha = 0.5$ or $\alpha = 0.75$.

Experimental series Using the various experimental parameters described in the preceding paragraphs, we perform a series of experiments:

Manual versus automatic ring design: Considering only IR path design, we vary the architecture (ring / mesh), the number of links available (14 / 25) and the ring design method (manual / automatic).

Break-even cost factor: Considering only the mesh architecture, we vary the number of links available (14 / 25) and the network element cost factor α, to determine when the total network costs are the same for ring and mesh architectures.

Integrated routing versus prerouting: Considering only ring architecture, we vary the ring design method (manual / automatic), the number of links available (14 / 25) and the path design method (IR / PR).

3.4.3 Results & discussion

The experiments were performed with the LP / ILP optimiser CPLEX 6.61 (ILOG Cooperation, 1999) on a 440MHz HP Series 9000 Model J7000 with 1Gb RAM per optimisation process, each experiment running for 1–20 minutes. We note that only the barrier solver of CPLEX was able to cope with the larger problems (IR path design with 25 links); all the other optimisation methods would either run out of memory or would not finish within a 2 hour limit.

Manual versus automatic ring design The results from the first set of experiments are shown in Table 3.6, with the integer-rounded values in Table 3.7. Note that in both tables, the cost has been distributed out onto the various network components: links, transit equipment (line cards), I/O equipment (tributary cards) and cores. The "*% Cost*" column gives the total cost relative to that of the manual ring design with optimal routing.

Arch	*Links*	*Ring*	——— *Cost* ——						*Rings*	*Nodes/Ring*	
		design	*Link*	*Transit*	*I/O*	*Core*	*Total*	*%*		*avg.*	*std. dev.*
ring	14	manual	1.21	1.32	0.36	0.45	3.34	*100.0*	3	5.33	0.82
		auto	1.09	1.18	0.33	0.41	3.02	*90.4*	6	7.50	6.10
	25	auto	0.68	0.70	0.35	0.24	1.97	*59.0*	20	4.20	6.60
mesh	14	—	0.54	2.52	—	0.65	3.71	*111.1*	—	—	—
	25	—	0.39	2.03	—	0.52	2.94	*88.0*	—	—	—

Table 3.6: Optimal LP solutions. The price unit is the cost of one thousand MS-SPRing cores

Arch	*Links*	*Ring*	——— *Cost* ——						*Rings*	*Nodes/Ring*	
		design	*Link*	*Transit*	*I/O*	*Core*	*Total*	*%*		*avg.*	*std. dev.*
ring	14	manual	1.20	1.32	0.36	0.46	3.35	*100.3*	3	5.33	0.82
		auto	1.09	1.18	0.35	0.42	3.04	*91.0*	6	7.50	6.10
	25	auto	0.68	0.70	0.35	0.29	2.02	*60.5*	20	4.20	6.60
mesh	14	—	0.54	2.53	—	0.73	3.81	*114.1*	—	—	—
	25	—	0.39	2.04	—	0.58	3.01	*90.1*	—	—	—

Table 3.7: Optimal LP solutions rounded up to integral values. The price unit is the cost of one thousand MS-SPRing cores

We observe that rounding up does not change the figures significantly, which is due to the fact that the demand volumes are large in relation to the unit size of the network elements. From the figures we further find the most cost-effective network is the automatically designed ring-network, using 25 links. This result is somewhat surprising, since one would normally expect a better utilisation of the network using the mesh design (Wuttisittikulkij et al., 2000). The reason can clearly be seen in the cost of the node equipment which is the dominating cost for the mesh network. Even though the savings on the links are substantial in the mesh case—up to 75%—the mesh solution is still more expensive than the ring solution. Notice further that using many rings pays off: The automatic solutions

contain 6 and 20 rings. Hence, the LP solver achieves significant savings, up to 40%, by using the appropriate rings and routing the demands optimally on these rings.

Break-even cost factor for mesh node components Perhaps surprisingly, the results from the preceding section indicate that a ring based network is cheaper to establish than one based on the protected mesh architecture—the reason being that the node costs (i.e., core and transit costs) are significantly higher in the mesh case.

Given a solution $(\vec{z}^{\mathrm{X}}, \vec{z}^{\mathrm{C}}, \vec{u})$ to a mesh network optimisation problem, a rough approximation of the cost of a mesh network where node costs are multiplied by a factor α is

$$\begin{aligned} cost^{\text{total}}_{\text{mesh}}(\alpha) &= \sum_n \alpha C^{\text{switching}} \cdot z^{\mathrm{X}}_n + \sum_n \alpha C^{\text{core}} \cdot z^{\mathrm{C}}_n + \sum_l C^{\text{link}}_l \cdot u_l \\ &= \alpha \cdot (cost^{\text{transit}}_{\text{mesh}} + cost^{\text{core}}_{\text{mesh}}) + cost^{\text{link}}_{\text{mesh}}. \end{aligned}$$

Thus, an approximate break-even factor can be determined by solving an equation for α:

$$\begin{aligned} & cost^{\text{total}}_{\text{ring}} = cost^{\text{total}}_{\text{mesh}}(\alpha) = \alpha \cdot (cost^{\text{transit}}_{\text{mesh}} + cost^{\text{core}}_{\text{mesh}}) + cost^{\text{link}}_{\text{mesh}} \\ \Updownarrow \\ & \alpha = \frac{cost^{\text{total}}_{\text{ring}} - cost^{\text{link}}_{\text{mesh}}}{cost^{\text{transit}}_{\text{mesh}} + cost^{\text{core}}_{\text{mesh}}} \end{aligned}$$

so we find $\alpha_{25\text{links}} = \frac{2.02-0.39}{2.04+0.58} \approx 0.6$ and $\alpha_{14\text{links}} = \frac{3.04-0.54}{2.53-0.73} \approx 0.8$. However, the changed node costs may influence the optimal solution, so we must perform some experiments where we vary α around these values, the results of which are shown in Table 3.8 on the following page. By comparing with the costs for ring networks given in Table 3.7 on the preceding page (3.04 using 14 links, and 2.02 using 25 links), it can be seen that protected mesh networks indeed are competitive when $\alpha \leq 0.6$.

Integrated routing versus prerouting of traffic demands The effect of integrating the optimisation of the traffic routing with the ring selection can be seen in Table 3.9 on the following page. It is interesting to note that there is a 10–15% loss because of the prerouting.

3.4.4 Conclusion

In this work we have investigated the costs for optimised protected networks. We have compared path protected mesh networks and ring networks using the

Links	α	*Link cost*	*Transit cost*	*Core cost*	*Total cost*	*%*
14	0.85	0.54	2.16	0.62	3.32	*99.4*
	0.80	0.54	2.03	0.58	3.15	*94.3*
	0.75	0.54	1.90	0.55	2.99	*89.5*
25	0.65	0.39	1.33	0.38	2.10	*62.9*
	0.60	0.39	1.22	0.35	1.97	*58.9*
	0.55	0.39	1.12	0.32	1.84	*55.1*

Table 3.8: Integer solutions for mesh network costs as a function of the mesh node cost factor α.

Links	*Ring*	*Path*	———*Cost* ——						*Rings*	*Nodes/Ring*	
	design	*design*	*Link*	*Transit*	*I/O*	*Core*	*Total*	*%*		*avg.*	*std. dev.*
14	manual	IR	1.21	1.32	0.36	0.46	3.35	*100.3*	3	5.33	0.82
		PR	1.45	1.53	0.26	0.53	3.77	*112.8*	3	5.33	0.82
	auto	IR	1.09	1.18	0.35	0.42	3.04	*91.0*	6	7.50	6.10
		PR	1.36	1.45	0.28	0.50	3.60	*107.8*	5	7.20	5.90
25	auto	IR	0.68	0.70	0.35	0.29	2.02	*60.5*	20	4.20	6.60
		PR	0.93	0.91	0.29	0.32	2.44	*73.1*	18	4.00	4.20

Table 3.9: The effect of the path design (IR / PR) on ring network costs, LP solution rounded up

same links, and surprisingly the ring networks are more than 25% cheaper. The cost difference is solely due to the higher cost of the switches used in the path protected solution: If the prices for these switches were reduced by a factor of 0.6 they would be competitive. Further, the routing of the demands plays a significant role in the costs: Simply using a standard shortest path routing followed by optimal ring selection results in a loss of 10 to 15%. Finally, it is not trivial to design ring networks: A manual design was improved more than 40% by the optimal LP design.

The conclusions drawn in this work are based on only one network, the backbone network of TDC Tele Danmark. They are, however, so significant that TDC Tele Danmark will consider re-evaluating the way the ring networks are designed, with an aim to reducing costs. Obviously it would be interesting to apply the presented design methods to other networks, but unfortunately the confidentiality of detailed information about backbone networks of telecommunication operators

has prevented us from doing this.

More investigations should be carried out to determine exactly the network size limits for using this LP-based ring design approach. For networks that exceed this limit several modifications could be applied to obtain approximations to the optimal design. Since the number of potential rings is the fastest growing factor of the problem, it seems reasonable to apply some heuristic to reduce the number of rings actually considered in the linear program. By looking at the solutions derived in this work, it seems that the smallest rings containing each demand are preferred; smaller capacities on the links might change this, though. Another possibility could be to reduce the number of demands by dividing the network into two or more parts, optimising each part individually and only exchanging aggregate demand information at the dividing nodes—this would also reduce the number of rings considered. Finally, for very large networks good heuristics will surely be needed (Wuttisittikulkij et al., 2000).

Part III

Dynamic Teletraffic

Chapter 4

MPλS networks

In this chapter we consider MPλS networks and report on some initial experiments with a network simulator that focus specially on the effect of delays and wavelength reservation strategies during setup and teardown of connections.

4.1 MPλS network model

The MPλS network model we consider consists of WDM nodes and links. Each node is equipped with an OXC which can perform strictly nonblocking switching. Each link contains one fibre carrying W wavelengths, and no wavelength conversion is possible in the nodes, so the wavelength continuity constraint must be satisfied. The average link delay—i.e., the link transmission time—on link l is denoted t_l^{link} and the node delay—i.e., the time it takes to process a network protocol message, including the time it takes to receive it—is denoted t_n^{node}.

As we are only concerned with the path setup and message forwarding protocol we so not consider network failures in the following, and implicitly assume given appropriate routing protocols, including link state messaging.

We consider static routing, so all network control messages carry information on the path along which they should be forwarded. Sun et al. (2001) use a two-stage MPλS path setup protocol where a probing message is sent downstream from source to destination, recording along the way which wavelengths are unused. The destination node then selects one of the unused wavelengths and issues a reservation message upstream. However, in this work we use the "forward reservation protocol with immediate release" proposed by Saha (2000), in which the wavelengths are reserved during downstream control message propagation, and any wavelengths blocked at intermediate links are immediately released by

issuing release messages upstream from the blocking link. Our protocol is thus based on three types of messages, all parameterised by a set of wavelengths ω:

Reserve ω**:** A request to reserve wavelengths, based on ω.

Release ω**:** A message that previously reserved wavelengths in ω are no longer needed.

Connected, release ω**:** A message signalling that the requested path has been set up, and that any reserved wavelengths in ω are no longer needed.

For simplicity we can assume that the messages are signalled between nodes using out-of-band signalling. An example of how the MPλS path setup protocol works for a path with links $1, \ldots, N-1$ is shown in Figure 4.1 on the next page, where the node and link delays have been exaggerated relative to the holding time. The set ω_n contains the wavelengths reserved by node n, and ω'_n contains the wavelengths released by node $n+1$. The sets ω_n and ω'_n are determined by the wavelengths that are free on the appropriate links and the reservation policy used. Letting $\text{free}(l)$ denote the set of free wavelengths on link l, we consider the following reservation policies:

Single wavelength reservation: Under this policy $\omega_1 = \{\lambda\}$, so just a single wavelength is reserved. If it is in use on any of the links along the path, the request is blocked.

Greedy wavelength reservation: Under this policy $\omega_1 = \text{free}(1)$ and $\omega_n = \omega_{n-1} \cap \text{free}(n)$, thus reserving as many wavelengths as possible for this path.

***k*-bounded wavelength reservation:** Under this policy $\omega_1 \subseteq \text{free}(1) \wedge |\omega_1| \leq k$ and $\omega_n = \omega_{n-1} \cap \text{free}(n)$, so at most k wavelengths are reserved on each link for this path.

When wavelengths can be released one can consider (at least) these policies:

Stepwise wavelength release: Under this policy $\omega'_n = \omega_n \setminus \omega_{n+1}$, so that wavelengths are released as soon as it is detected that they cannot be used for the path which is being set up.

Final-step wavelength release: Under this policy, which is a generalisation of the two preceding policies, $\omega'_1 = \omega'_2 = \cdots = \omega'_{n-3} = \{\}$ and $\omega'_{n-2} = \omega_1 \setminus \omega_{n-1}$, so that reserved wavelengths are first released when the entire path has been set up, thus saving signalling bandwidth.

In the following, however, we will just consider the stepwise wavelength release policy.

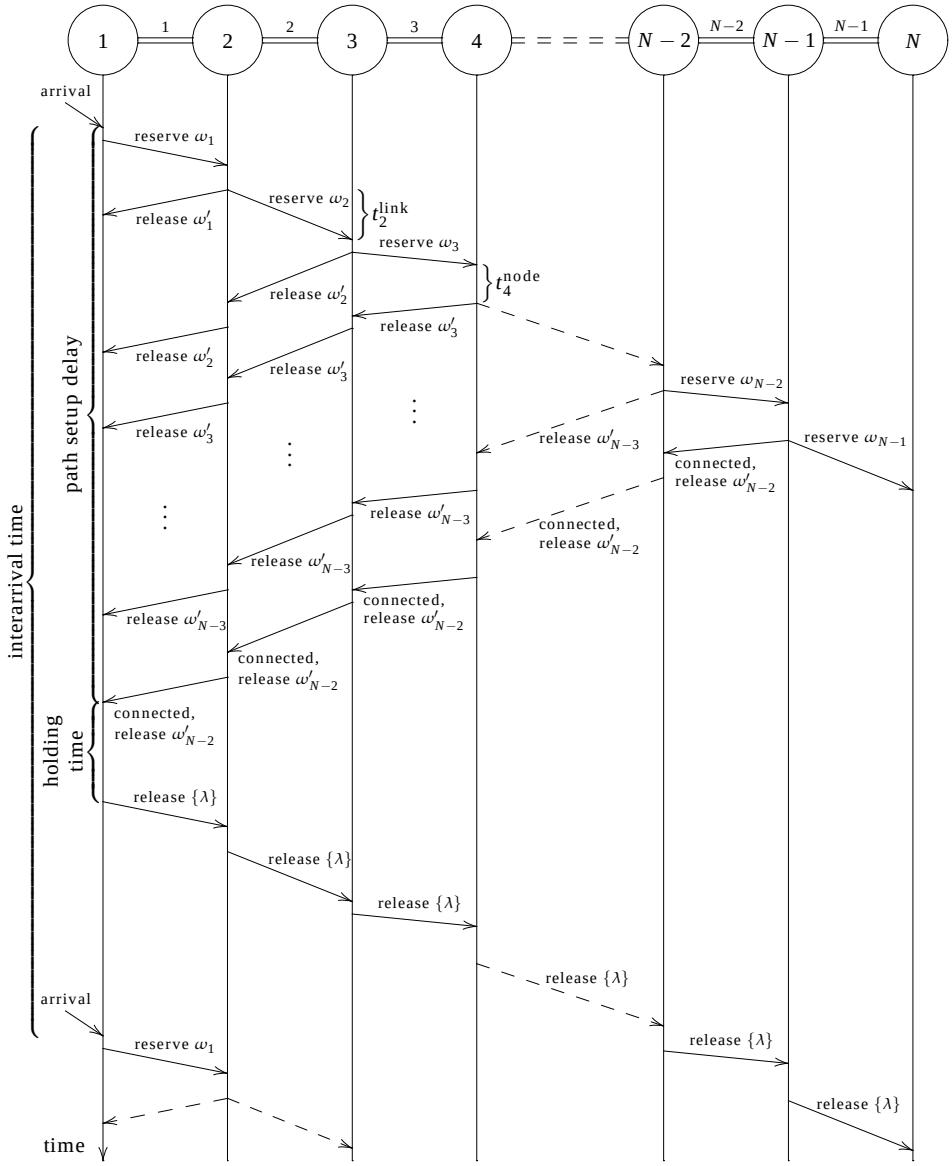


Figure 4.1: Example of MPλS path setup from node 1 to node N

4.2 MPλS network simulator

We construct an MPλS network simulator which is able to simulate a circuit-switched loss network with the setup and teardown protocols described in Section 4.1 for arbitrary network topologies.

Input to the simulator is comprised of:

Topology: A set of nodes and a set of links, along with the delay in each node or link, and the number of wavelengths per fibre.

Traffic: A set of demands, where each demand is assigned an arrival process and a fixed route. The arrival process is characterised by its type (Poisson, Binomial, Pascal or Constant), its average interarrival time $1/\lambda$ and average holding time $1/\mu$.

Strategy: Wavelength reservation and release policies.

Output from the simulator consists of:

Blocking: For each demand, the call blocking probability, that is, the number of accepted calls divided by the total number of call attempts.

When a link receives a request for reserving k wavelengths of a set ω of wavelengths, it examines the free wavelengths on the link in a locally cyclical manner and returns the first k free wavelengths that are also in ω—or fewer if this is not possible. The next request to the link will start searching from the wavelength following the last considered wavelength.

The simulator engine is based on the event advance design (Perros, 1999). As the blocking probabilities observed with the simulator in general are correlated, we use the following procedure to estimate the confidence intervals: Call attempts for each demand are observed in batches, the size of which can be set by the user. The simulator continues until the confidence interval is below a given threshold and at least 31 batches have been gathered for each demand, discarding all first batches to avoid data from the transient period. The 95% confidence interval is then computed as $\overline{\overline{B}} \pm 1.96\frac{s}{\sqrt{n}}$ where n is the number of batches, $\overline{\overline{B}} = \frac{1}{n}\sum_{i=1}^{k}\overline{B}_i$, $s^2 = \frac{1}{n-1}\sum_{i=1}^{k}(\overline{B}_i - \overline{\overline{B}})^2$ and $\overline{B}_i$ is the average blocking observed in batch i.

4.3 Network and traffic examples

We run the simulator on two networks: The 10-link linear network shown in Figure 4.2, and the 11-link mesh network shown in Figure 4.3. In the 11-link network

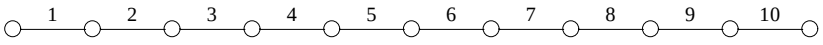


Figure 4.2: Linear 10-link network example

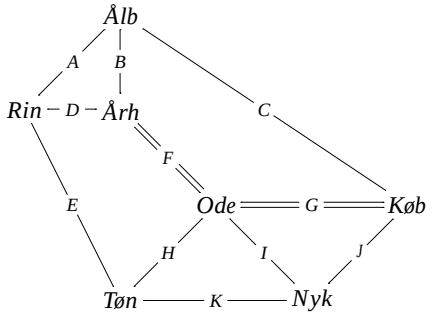


Figure 4.3: 11-link mesh network example

there are two fibres between *Årh* and *Ode*, and two fibres between *Ode* and *Køb*, as these three cities have the highest traffic demands, due to their population sizes.

4.3.1 Traffic for the 10-link linear network

We use three kinds of traffic which create different offered link load patterns: mountain, uniform and bowl traffic. In the following we let $2L = 10$, the number of links in the network.

Mountain (negative parabola) traffic

The arrival intensities λ for the demands in this pattern are calculated by letting the intensity for all demands be equal, resulting in an offered load for link number

l of $c(l \cdot (2L+1-l)) = c(-l^2 + (2L+1)l)$. For $c = 0.1$ we have

last hop in path \ *first hop in path*	1	2	3	4	5	6	7	8	9	10
1	0.10000									
2	0.10000	0.10000								
3	0.10000	0.10000	0.10000							
4	0.10000	0.10000	0.10000	0.10000						
5	0.10000	0.10000	0.10000	0.10000	0.10000					
6	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000				
7	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000			
8	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000		
9	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	
10	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000	0.10000

The resulting offered link loads for a holding time of $\frac{1}{\mu} = 1$ are shown in Figure 4.4.

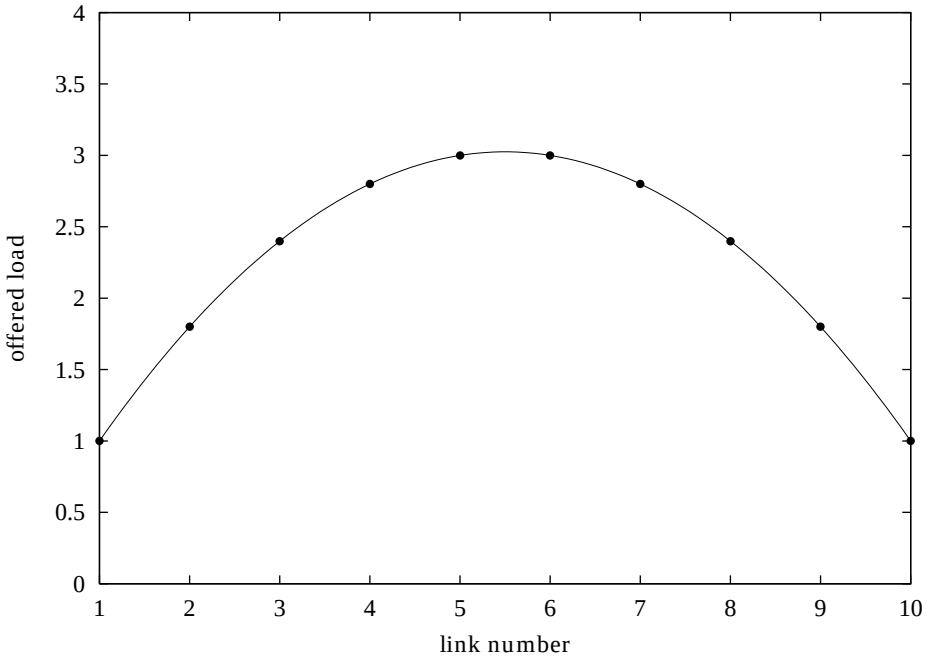


Figure 4.4: Mountain traffic for the 10-link linear network

Link load uniform traffic

The arrival intensities for the demands in this pattern give the same offered load on all links, by letting the intensity of a demand be proportional to $2/(x^2 + x)$, where $x = (L - d)$ and d is how close to the middle links the demand path is. The offered demand loads are as follows:

last hop in path \ *first hop in path*	1	2	3	4	5	6	7	8	9	10
1	1.00000									
2	0.33333	0.33333								
3	0.16667	0.16667	0.16667							
4	0.10000	0.10000	0.10000	0.10000						
5	0.06667	0.06667	0.06667	0.06667	0.06667					
6	0.06667	0.06667	0.06667	0.06667	0.06667	0.06667				
7	0.06667	0.06667	0.06667	0.06667	0.06667	0.06667	0.10000			
8	0.06667	0.06667	0.06667	0.06667	0.06667	0.06667	0.10000	0.16667		
9	0.06667	0.06667	0.06667	0.06667	0.06667	0.06667	0.10000	0.16667	0.33333	
10	0.06667	0.06667	0.06667	0.06667	0.06667	0.06667	0.10000	0.16667	0.33333	1.00000

The resulting offered link loads for a holding time of $\frac{1}{\mu} = 1$ are shown in Figure 4.5 on the next page.

Bowl (positive parabola) traffic

The loads in this pattern give a parabola with positive coefficient, resulting in a load for link number l of $c(l \cdot (l - 2L - 1) + L \cdot (L + 1)) + 1 = c(l^2 - (2L + 1)l + (L^2 + L)) + 1$. For $2L = 10$ and $c = 0.1$ we have and are as follows:

last hop in path \ *first hop in path*	1	2	3	4	5	6	7	8	9	10
1	1.90000									
2	0.56667	0.56667								
3	0.23333	0.23333	0.23333							
4	0.10000	0.10000	0.10000	0.10000						
5	0.03333	0.03333	0.03333	0.03333	0.03333					
6	0.03333	0.03333	0.03333	0.03333	0.03333	0.03333				
7	0.03333	0.03333	0.03333	0.03333	0.03333	0.03333	0.10000			
8	0.03333	0.03333	0.03333	0.03333	0.03333	0.03333	0.10000	0.23333		
9	0.03333	0.03333	0.03333	0.03333	0.03333	0.03333	0.10000	0.23333	0.56667	
10	0.03333	0.03333	0.03333	0.03333	0.03333	0.03333	0.10000	0.23333	0.56667	1.90000

The resulting link loads are shown in Figure 4.6 on page 158.

4.3.2 Traffic for the 11-link mesh network

The traffic used for the 11-link mesh network is shown in the upper triangular half of Table 4.1 on page 159, and the routing of each demand is shown in the

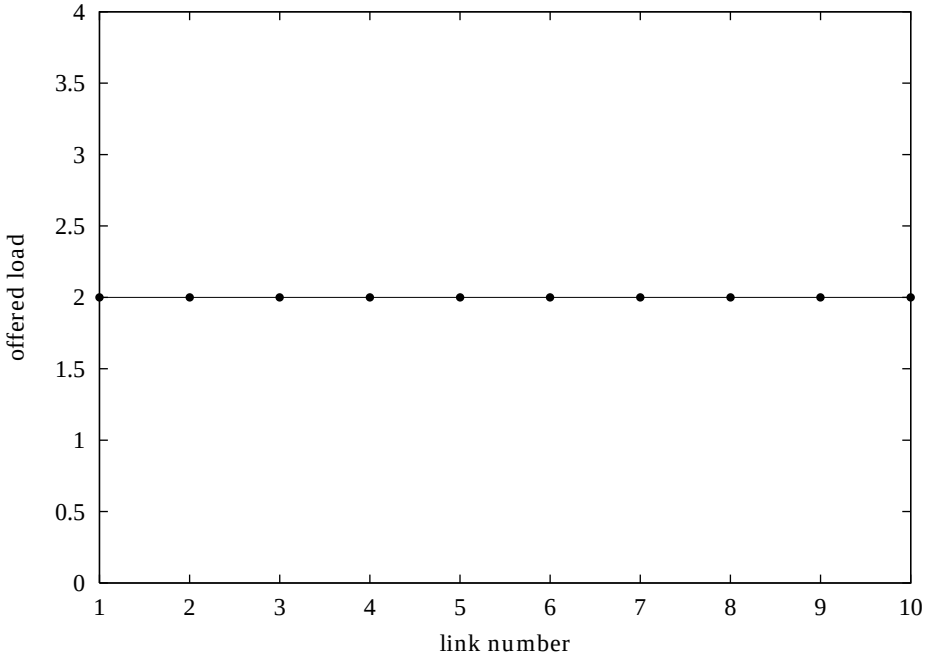


Figure 4.5: Uniform traffic for the 10-link linear network

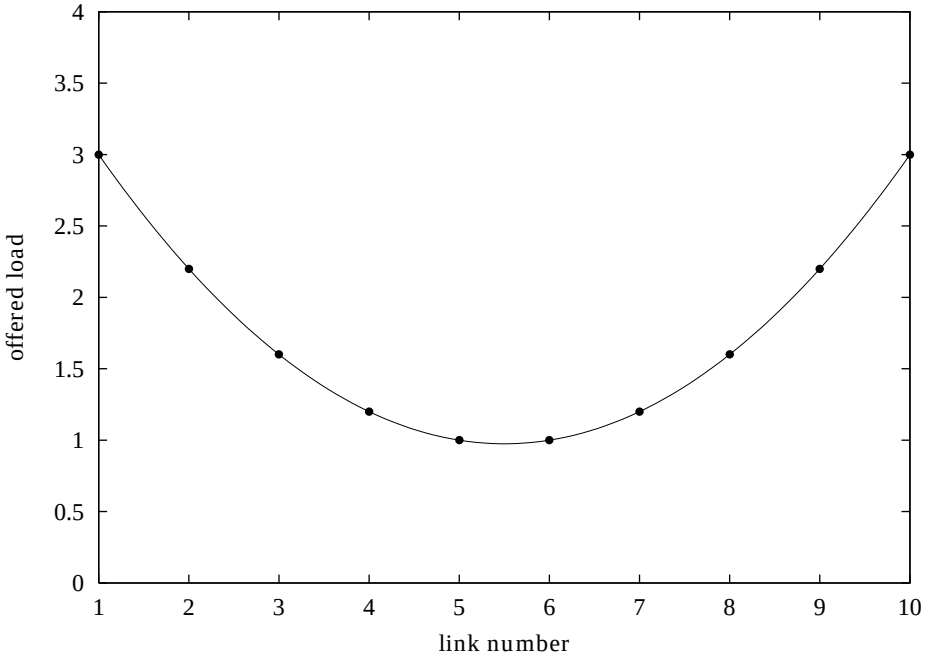


Figure 4.6: Bowl traffic for the 10-link linear network

	Ålb	*Rin*	*Årh*	*Ode*	*Køb*	*Tøn*	*Nyk*
Ålb		4	24	16	28	8	12
Rin	*DB*		6	4	7	2	3
Årh	*B*	*D*		24	42	12	18
Ode	*HEA*	*HE*	F_1		28	8	12
Køb	*C*	*CA*	G_2F_2	G_1		14	21
Tøn	*EA*	*E*	*ED*	*H*	*KJ*		6
Nyk	*JC*	*KE*	IF_1	*I*	*J*	*K*	

link	*traffic*
A	31
B	28
C	47
D	22
E	45
F_1	42
F_2	42
G_1	28
G_2	42
H	28
I	30
J	37
K	23

Table 4.1: Traffic and routing, and offered link traffic for the 11-link mesh network example

lower half of the table. The offered link traffic for each link, assuming an average holding time of $\frac{1}{\mu} = 1$ is also shown.

4.4 Empirical results

We run the simulator with the network and traffic examples given in the preceding text using Poisson arrivals, letting node delay, link delay and holding time distribution be the negative exponential distribution, and using $W = 32$ wavelengths.

The batch size is set to 3000 call attempts and the confidence interval threshold to 0.02.

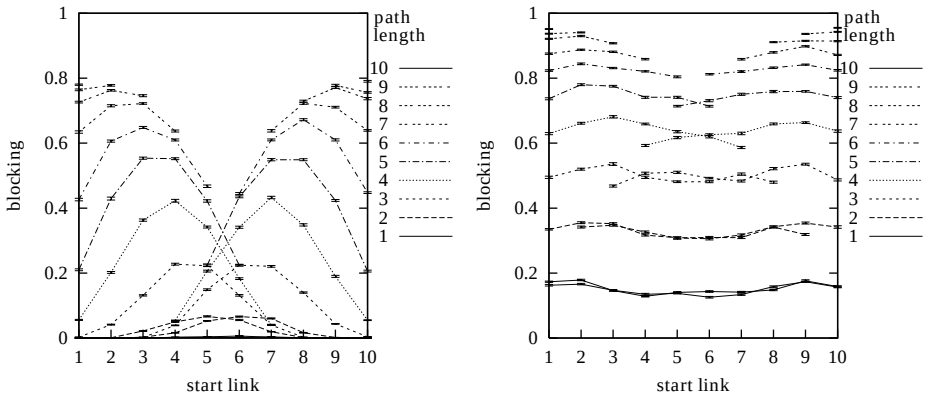
4.4.1 Blocking in the 10-node linear network

For the 10-link linear network we let the average holding time be $\frac{1}{\mu} = 8$, and for each of the 55 paths we use the given mountain, uniform and bowl arrival intensities *for both directions.* Thus, we find that a link in the uniform traffic pattern has an offered traffic per wavelength of $2\lambda\frac{1}{\mu}/W = 2 \cdot 2 \cdot 8/32 = 1$ erlang.

The effect of network delays

In this section we shall only consider the greedy reservation strategy—i.e., reserving as many wavelengths as possible—and stepwise wavelength release.

Equal link and node delays Setting the average node and link delay equal and varying it 0–0.1–1 time units, and varying the traffic pattern mountain–uniform–bowl we observe the simulated blocking probabilities shown in Figures 4.7–4.10. The graphs starting from the left side of the diagrams represent demands from



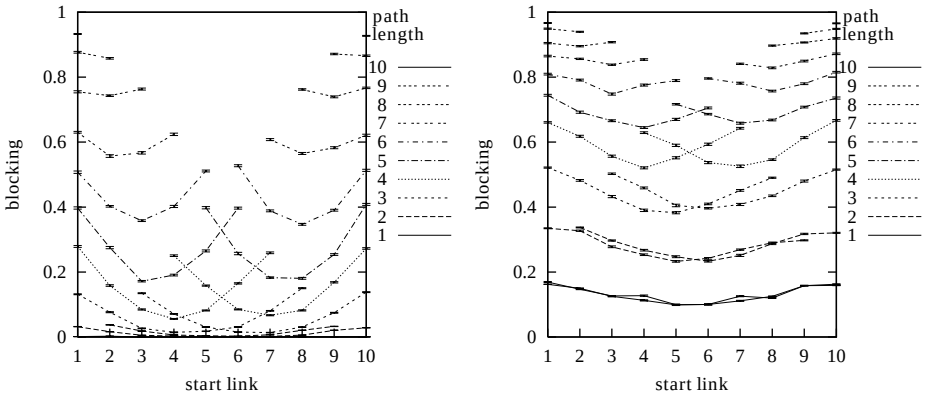
(a): 0 link delay, 0 node delay (b): 0.1 link delay, 0.1 node delay

Figure 4.7: Results for mountain traffic

lower to higher numbered hops, and vice versa; the 95% confidence intervals are indicated for each observation by two small horizontal lines.

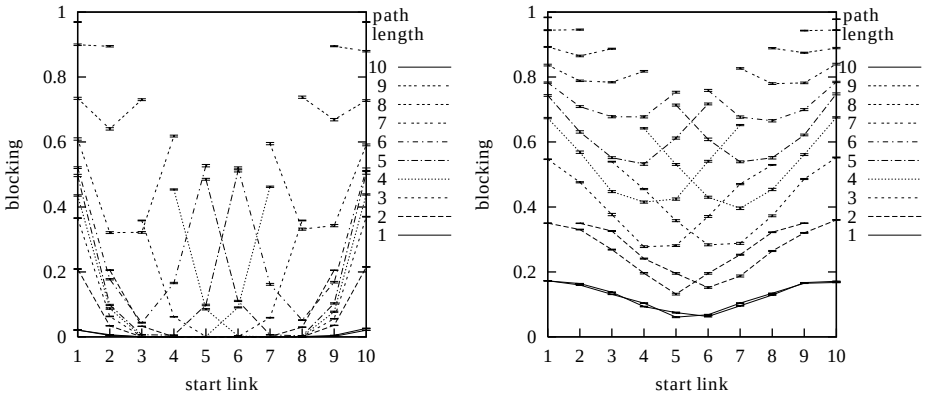
In Figure 4.7a we can see that when there is no setup delay and the traffic demands are of similar sizes, the blocking of a call is chiefly determined by the offered loads on the links that its path use. If some demands are offered greater traffic than others, they will dominate, as seen in Figures 4.8a and 4.9a; for bowl traffic, the single hop demands are up to 57 times larger than the smallest traffic demand, so the blocking of demands for the longer paths are mainly determined by the offered loads on the single-hop paths. Even for link load uniform traffic, this effect is seen, cf. Figure 4.8a.

When there is a setup delay, the effect of the offered link loads is much reduced, and the blocking of a demand depends more on the number of hops in its path, cf. Figure 4.7b, although it is still somewhat dependent on the offered



(a): 0 link delay, 0 node delay (b): 0.1 link delay, 0.1 node delay

Figure 4.8: Results for link load uniform traffic



(a): 0 link delay, 0 node delay (b): 0.1 link delay, 0.1 node delay

Figure 4.9: Results for bowl traffic

traffic of the largest demands, as seen with link load uniform and bowl traffic in Figures 4.8b and 4.9b.

Finally, when setup delay is large—for paths longer than 4 hops the average delay is greater than the average holding time—the blocking probabilities are almost only determined by the number of hops, cf. Figure 4.10 on the facing page: for uniform traffic the blocking of a path with h hops is approximately $1 - (1 - 0.45)^h$, regardless of the starting node.

This can be explained by considering that the wavelengths are blocked by reservations in a significant part of the time, so arrivals for demands with low offered traffic have the same chance of success as those for high offered traffic.

Link and node delay ratio Next, we fix the average link delay at 0.1 time units, and varying the traffic pattern mountain–uniform–bowl and node delays 0–0.1–1–10, we observe the simulated blocking probabilities shown in Figures 4.11–4.16. Comparing the (a) and (b) graphs in Figures 4.11–4.13 we see that node delays produce higher blocking probabilities than link delays of the same magnitude. When the node delays are large relative to the link delays, the blocking probabilities mainly depend on the number of hops in the paths, cf. Figures 4.14–4.16.

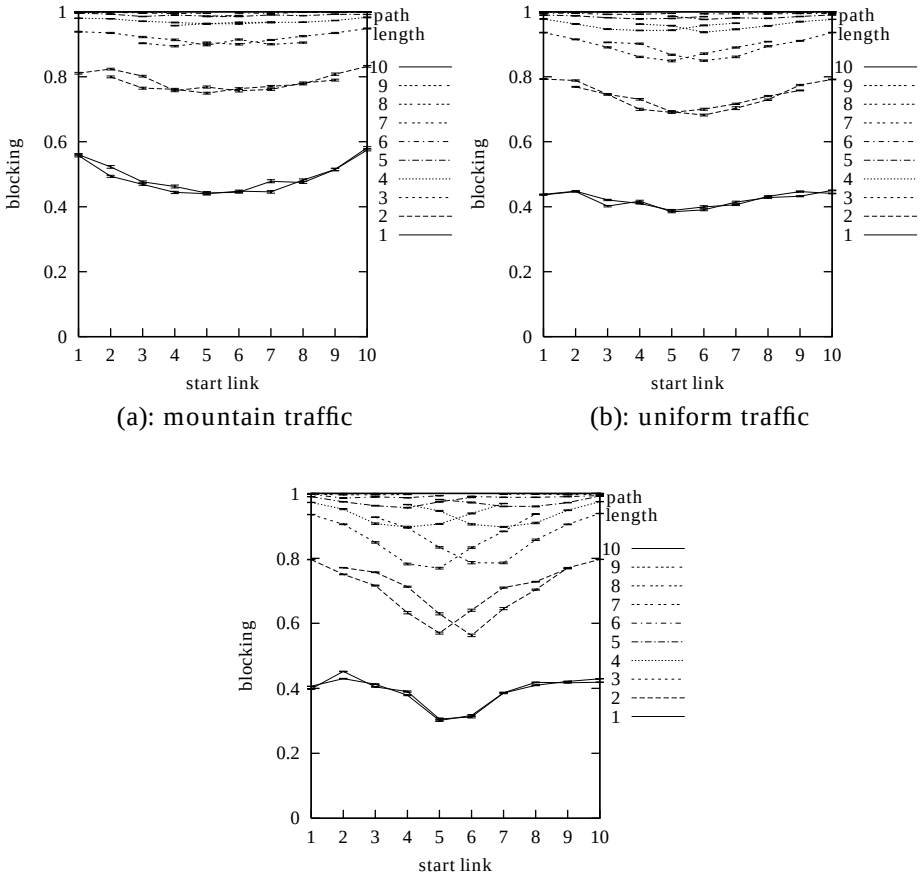
Reservation strategy

We turn now to the question of how the reservation strategy influences the blocking probabilities. Considering again the 10 node network and fixing the link and node delays to 0.1 and 1, respectively, the blocking probabilities using k-bounded reservation for $k = 1$, 2, 8, 16 and 32 are shown in Figures 4.17–4.22 and Figures 4.14a–4.16a.

We see that the blocking for one-hop paths is zero for single wavelength reservation, which is due to much fewer links being blocked by wavelength reservations, compared with k-bounded wavelength reservation for $k > 1$.

Furthermore, although the blocking probabilities are generally reduced when going from 1-bounded to 2-bounded wavelength reservation, they increase when stepping up to 8-bounded wavelength reservation.

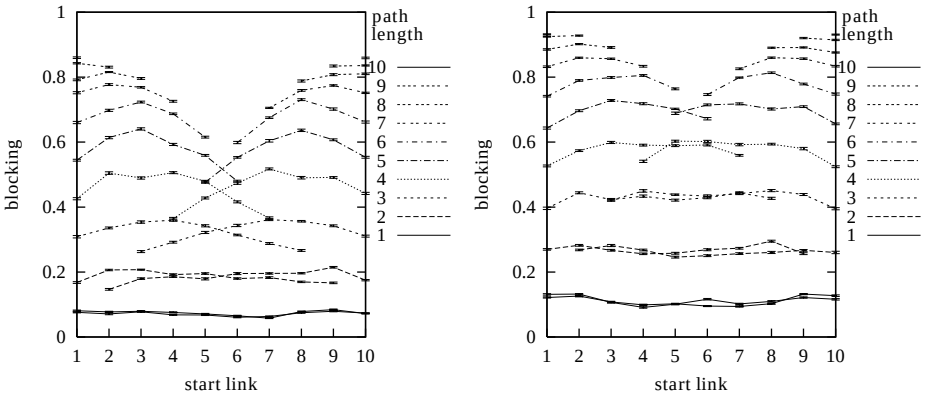
Finally, for small k the k-bounded wavelength reservation reveal a significant difference in the blocking probability of a path depending on which end it is set up from. Comparing the blocking probabilities in Figure 4.17a and 4.19a for the 2-hop path using link 1 and 2, we see that starting reservation at the highest loaded link results in the lowest blocking probability, which makes sense.



(a): mountain traffic (b): uniform traffic

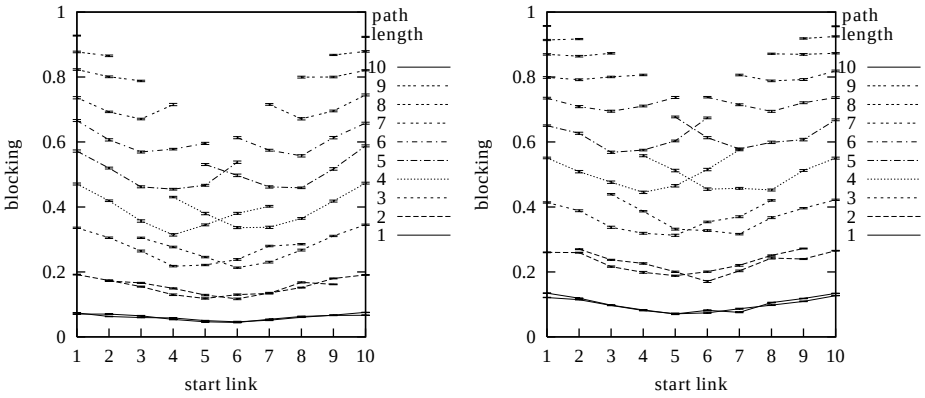
(c): bowl traffic

Figure 4.10: Results for 1 link delay, 1 node delay bowl traffic



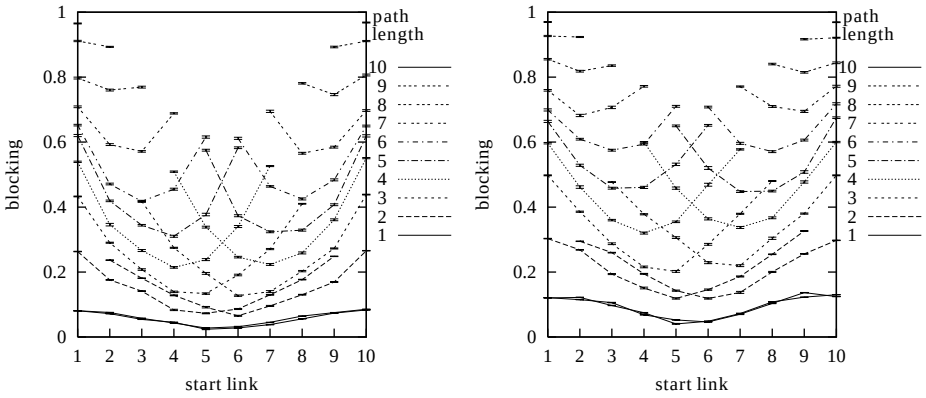
(a): 0.1 link delay, 0 node delay (b): 0 link delay, 0.1 node delay

Figure 4.11: Results for mountain traffic



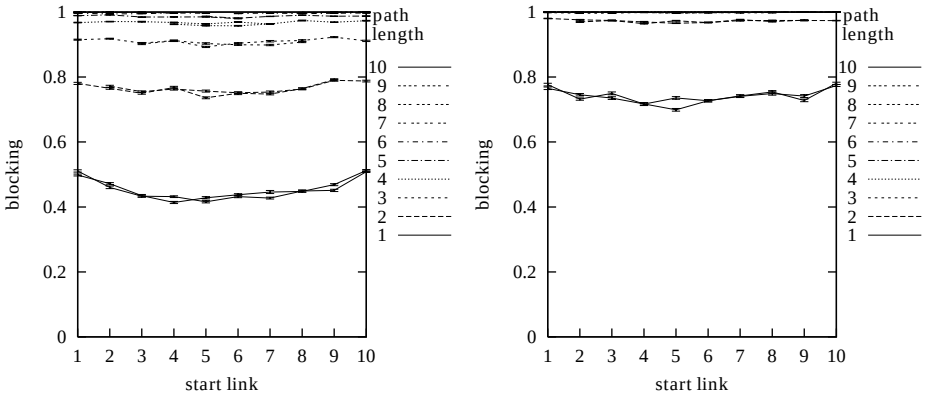
(a): 0.1 link delay, 0 node delay (b): 0 link delay, 0.1 node delay

Figure 4.12: Results for uniform traffic



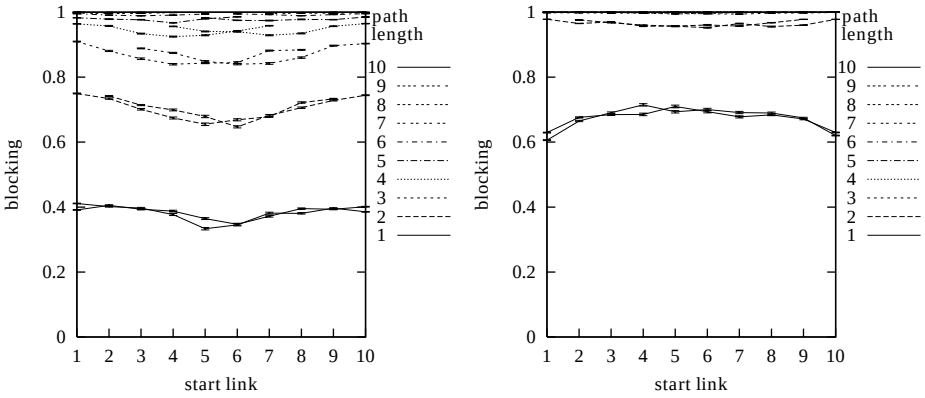
(a): 0.1 link delay, 0 node delay (b): 0 link delay, 0.1 node delay

Figure 4.13: Results for bowl traffic



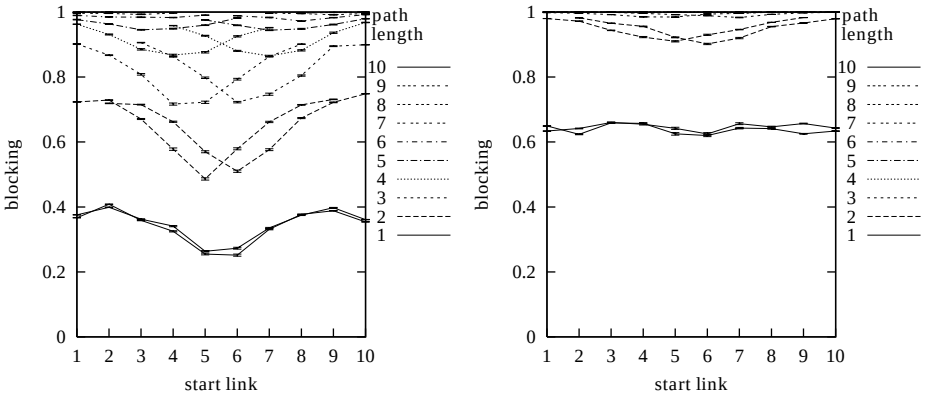
(a): 0.1 link delay, 1 node delay (b): 0.1 link delay, 10 node delay

Figure 4.14: Results for mountain traffic



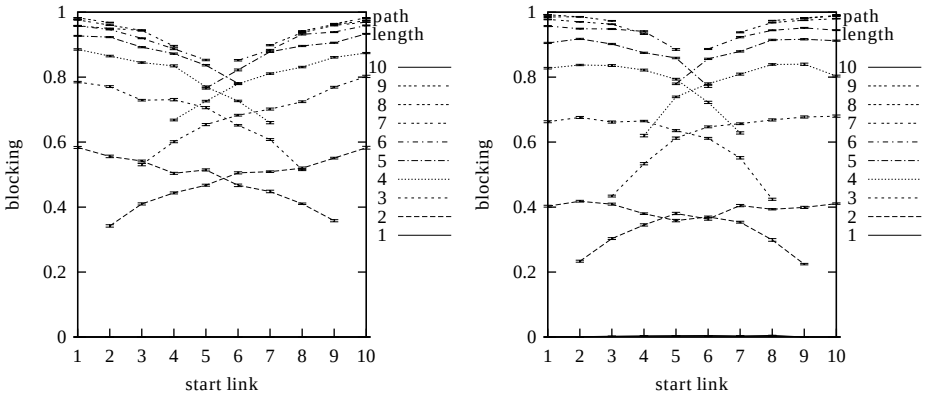
(a): 0.1 link delay, 1 node delay (b): 0.1 link delay, 10 node delay

Figure 4.15: Results for uniform traffic



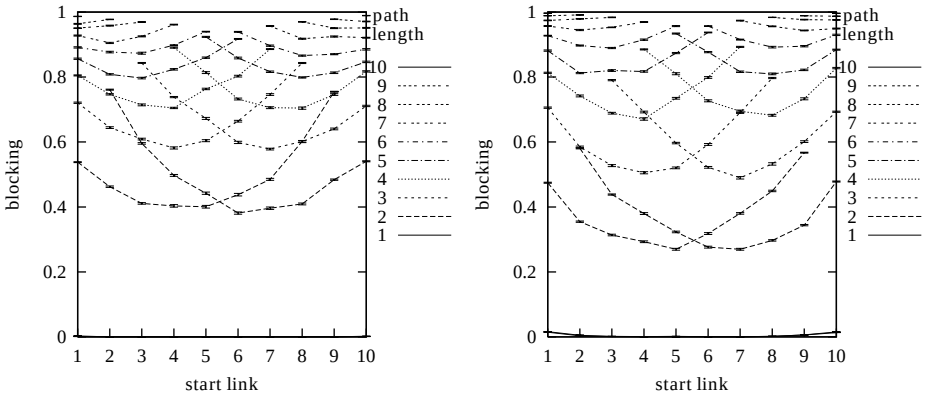
(a): 0.1 link delay, 1 node delay (b): 0.1 link delay, 10 node delay

Figure 4.16: Results for bowl traffic



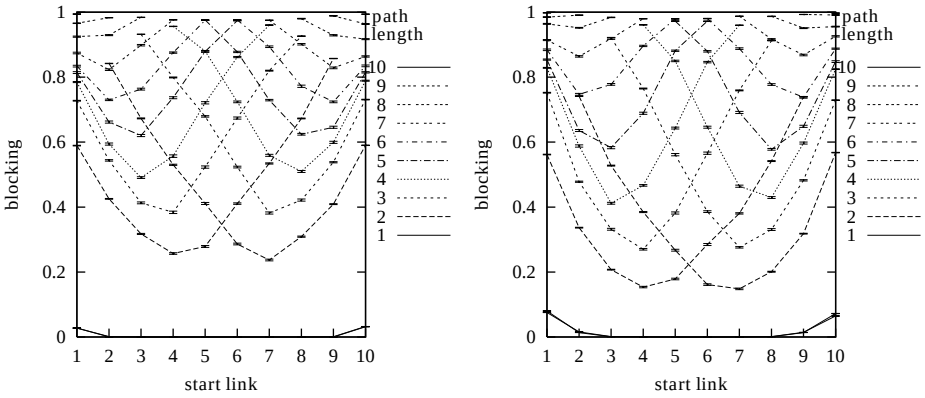
(a): 1-bounded wavelength reservation (b): 2-bounded wavelength reservation

Figure 4.17: Results for 0.1 link delay, 1 node delay mountain traffic



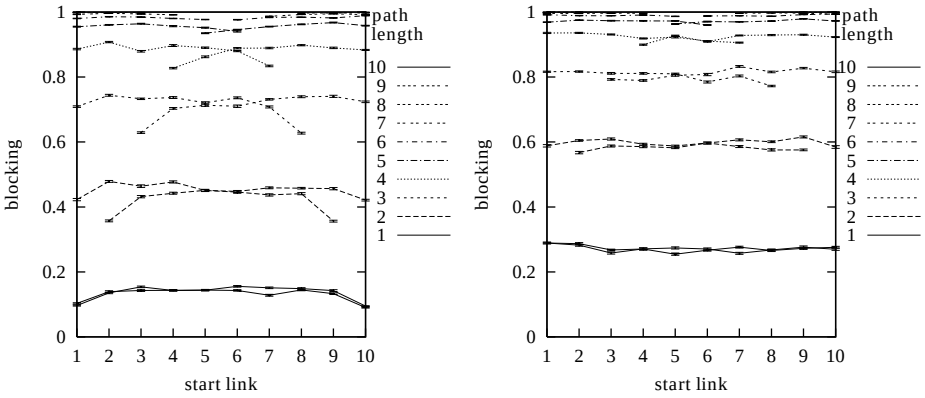
(a): 1-bounded wavelength reservation (b): 2-bounded wavelength reservation

Figure 4.18: Results for 0.1 link delay, 1 node delay uniform traffic



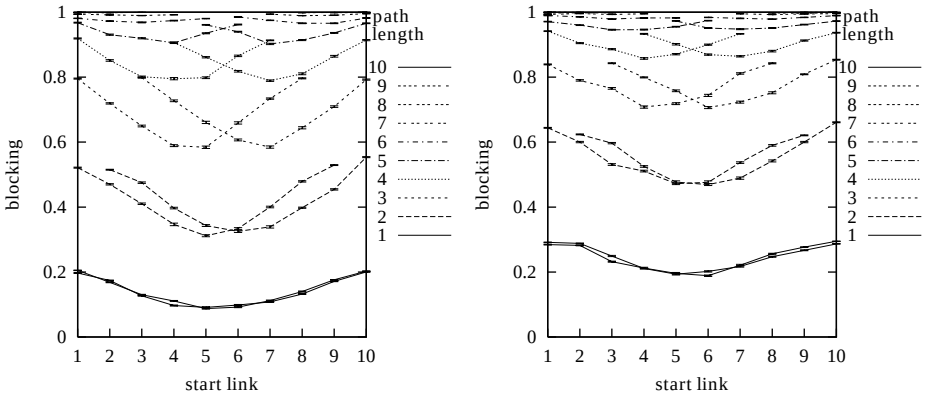
(a): 1-bounded wavelength reservation (b): 2-bounded wavelength reservation

Figure 4.19: Results for 0.1 link delay, 1 node delay bowl traffic



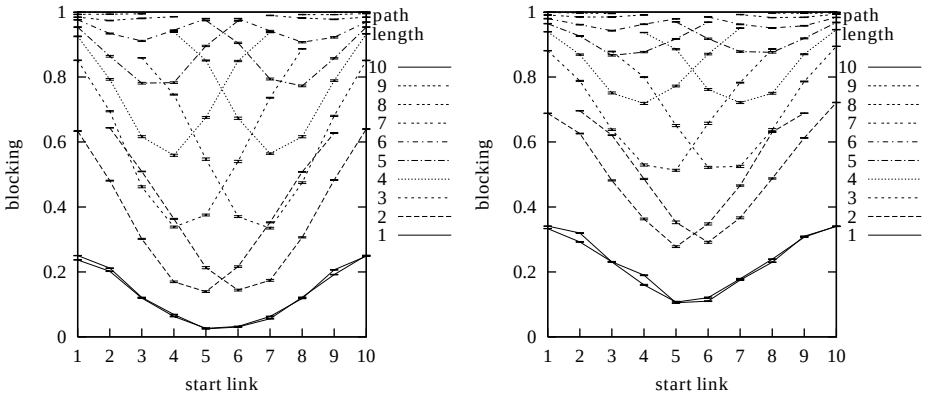
(a): 8-bounded wavelength reservation (b): 16-bounded wavelength reservation

Figure 4.20: Results for 0.1 link delay, 1 node delay mountain traffic



(a): 8-bounded wavelength reservation (b): 16-bounded wavelength reservation

Figure 4.21: Results for 0.1 link delay, 1 node delay uniform traffic



(a): 8-bounded wavelength reservation (b): 16-bounded wavelength reservation

Figure 4.22: Results for 0.1 link delay, 1 node delay bowl traffic

4.4.2 Blocking in the 11-link mesh network

For the 11-link mesh network we investigate what effects the node delays have on the blocking probabilities for the various demands using 32 wavelengths per fibre. We simulate Poisson arrivals with the intensities λ shown in Table 4.1 on page 159 and negative exponentially distributed holding times with mean $\frac{1}{\mu} = 2$. Fixing the link delay at 0.01 and varying the node delay 0.01–0.1–1–10–100 we obtain the results for single and for greedy wavelength reservation shown in Figures 4.23 and 4.24. Clearly, the longer paths have higher blocking probabilities

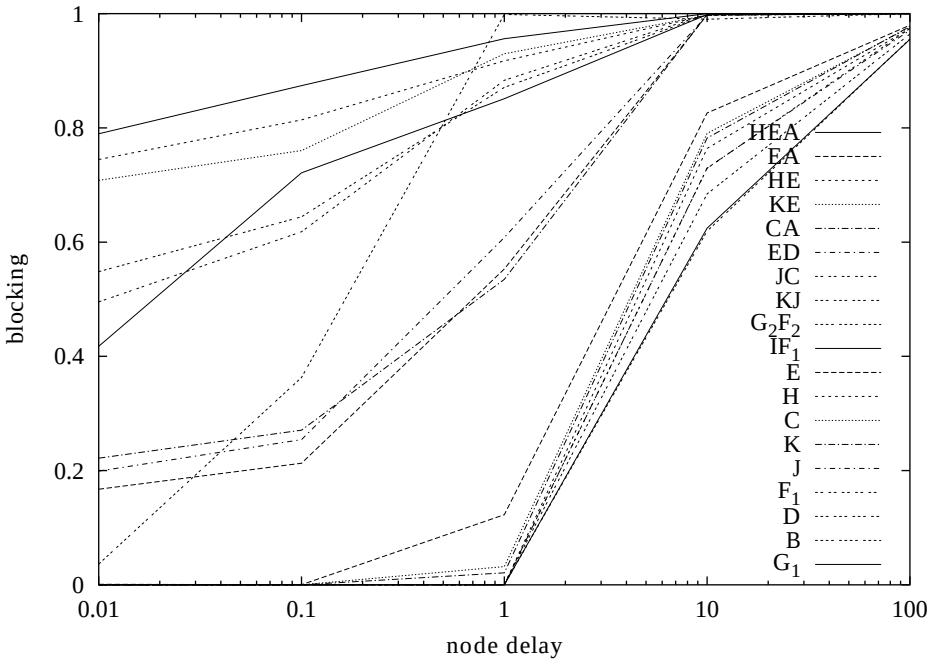


Figure 4.23: The effect of node delays on blocking probabilities, single wavelength reservation

and are more affected by increases in node delays.

Comparing the two figures we also note that greedy wavelength reservation reduces blocking probabilities for the longer routes—at least for low node delays—but at the price of increasing blocking dramatically on single-hop routes. This is to be expected, because when several wavelengths can be reserved on the

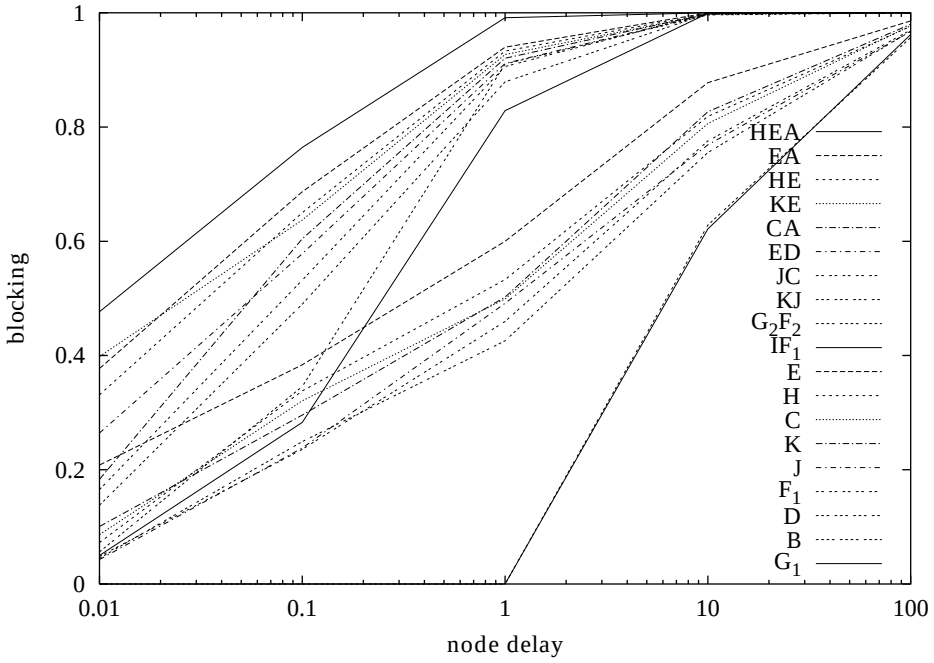


Figure 4.24: The effect of node delays on blocking probabilities, greedy wavelength reservation

first hop of a multihop path, the chances of finding a common free wavelength on the subsequent hops are better.

4.5 Conclusions

In this chapter we have reported on results from preliminary experiments performed with an MPλS network simulator constructed specifically to simulate the delays during setup and teardown of MPλS connections.

We find that when there is no delay and when traffic demands are of similar sizes, the call blocking is chiefly determined by the offered loads on the links the demand uses. When the setup delay is larger, however, the blocking is determined by the number of hops. We also find that node delays yield higher blocking probabilities than corresponding link delays.

Considering reservation strategy, the single wavelength reservation generally gives lower blocking probabilities, except for longer paths that might benefit from being able to reserve several wavelengths on the first hop of their paths.

Future work on MPλS setup delays could include modelling an optical burst switch scheme in which the setup delay is reduced by transferring data before connection success confirmation reaches the source node. Further, an analytical model for predicting blocking probabilities would alleviate the problem of long simulation times for networks with low intensity traffic sources.

Chapter 5

Analytic WDM blocking calculation

The added wavelength continuity constraint can increase the blocking in a circuit-switched WDM network, and it is important to estimate the size of the increase. In this chapter we present a way of using existing analytical network analysis tools to perform this estimation, changing only the link / route table which is passed to the analyser. Using solid theoretical definitions as a foundation, we transform the problem into a graph colouring problem, from which we then extract a minimised and sufficient set of *constraints*. These constraints make up a constraint / route table which is typically an extension of the link / route table.

The method is in principle able to handle networks of any size, as well as networks with multifibre links, but the complexity of the algorithms will in practice set an upper limit.

Assuming Poisson distributed arrival processes, we perform a systematic empirical study of networks with five, six and seven links, as well as a Danish national network.

5.1 Background

In the following section we will briefly review how blocking probabilities can be calculated from link / route tables.

5.1.1 Blocking models of WDM networks

Several models of blocking in WDM networks have been devised (Barry and Humblet, 1996; Birman, 1996; Kovačević and Acampora, 1995; Leith et al., 1999; Sridharan and Sivarajan, 2000; Tripathi and Sivarajan, 2000), typically including approximations (e.g., regarding link correlation) and iterations to reduce complexity. Most of these models use specific wavelength assignment schemes, and Ramaswami and Sivarajan (1995) have shown how to calculate a lower bound on the blocking under any scheme, using an integer linear program. In the work we present here we assume that an optimal wavelength assignment is used—which may entail re-assigning wavelengths to calls in progress. Furthermore, we assume given a static routing of the traffic demands.

5.1.2 Network analysis tools

Given a description of a statically routed conventional circuit-switched electrical network, tools exist for calculating the end-to-end blocking characteristics of it (Pinsky and Conway, 1992; Dickmeiss and Larsen, 1993; Iversen, 1987; Listov-Saabye and Iversen, 1989). The network description is given as a link / route table—an example is given in Table 5.1 on the following page that describes the simple network of Figure 5.1.

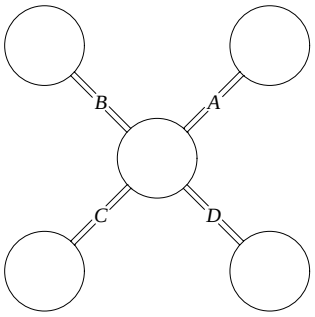


Figure 5.1: A four link network. Each network link holds $k = 2$ "wires."

5.1.3 Extending link/route tables

The key point of the method we present here is to *extend* the link / route table when analysing a WDM network. The rows representing the links can be regarded as *constraints:* the first line of Table 5.1 on the following page can be read as $A + AB +$

		Routes										
		A	*B*	*C*	*D*	*AB*	*AC*	*AD*	*BC*	*BD*	*CD*	*c*
Links	*A*	1				1	1	1				1
	B		1			1			1	1		1
	C			1			1		1		1	1
	D				1			1		1	1	1

Table 5.1: The link / route table for the electrically switched 4-link star network. c = capacity per wire

$AC + AD \leq 1 \cdot k$, where the variables A, AB, AC, AD count the number of calls in progress along the corresponding routes, and k is the number of wavelengths per link.

For the WDM network we add to Table 5.1 some extra constraints, resulting in Table 5.2. This extension of the link / route table now describes exactly what rout-

	Routes										
	A	*B*	*C*	*D*	*AB*	*AC*	*AD*	*BC*	*BD*	*CD*	*c*
	1				1	1	1				1
		1			1			1	1		1
			1			1		1		1	1
				1			1		1	1	1
					1	1		1			1
					1		1		1		1
						1	1			1	1
								1	1	1	1

Constraints

Table 5.2: Route constraint table for the all-optical WDM 4-link star network. c = capacity per wavelength

ing is possible in the WDM network, and when it is passed to the analysis tools, they calculate correctly the characteristics of the WDM network. In this way we are able to re-use all the existing link / route-based network analysis tools from electrically circuit-switched networks to analyse the WDM network characteristics.

5.2 Analysing WDM networks

We represent the WDM network using the following

Definition 4 (WDM Network) *A WDM network is a pair $N = (L, R)$, where L is a set of links and $R \subseteq \mathcal{P}(L)$ is a set of routes.*

5.2.1 Creating constraints

Given a WDM network, we can create a graph showing which routes interfere—i.e., share one or more links—by an interference graph (denoted G_p by Ramaswami and Sivarajan, 1995):

Definition 5 (Interference graph) *The interference graph for a WDM network (L, R) is an undirected graph $G = (R, E)$, where $E = \big\{\{q, r\} \mid \exists q, r \in R : q \neq r \wedge q \cap r \neq \emptyset\big\}$.*

An example of a 5 link, 6 route WDM network and the corresponding interference graph is shown in Figure 5.2. Note that to ease readability we have omitted the actual links and only shown the routes, as dotted lines.

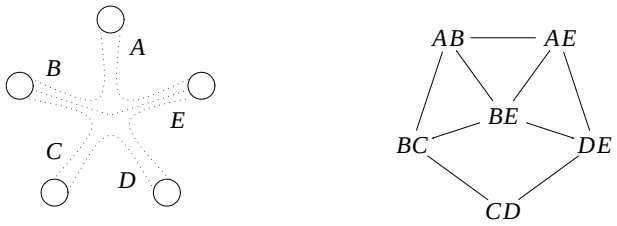


Figure 5.2: A WDM network (routes shown as dotted lines) and its corresponding interference graph

Definition 6 (Routing & colouring) *A routing ρ is a set of pairs $\{(r_1, i_1), \ldots, (r_n, i_n)\} \subseteq (R \times \mathbb{N})$, indicating for $k = 1, \ldots, n$ that i_k connections are routed along route r_k.*

Given a set of colours Λ, a colouring γ is a set of pairs $\{(r_1, \lambda_1), \ldots, (r_n, \lambda_n)\} \subseteq (R \times \Lambda)$.

We define the function routes $: (R \times \Lambda) \to (R \times \mathbb{N})$ *for counting the number of times a route is used:*

$$\text{routes}(\gamma) = \left\{ (r, i) \;\middle|\; \begin{array}{l} i > 0 \wedge \\ i = |\{(r', \lambda') \in \gamma \mid r' = r\}| \end{array} \right\}.$$

Thus, for instance, $\gamma_1 = \{(AB, red), (AE, green), (CD, red), (CD, green)\}$ is a colouring for just 2 colours, and $\text{routes}(\gamma_1) = \{(AB, 1), (AE, 1), (CD, 2)\}$ is the corresponding routing.

Suppose now that each link in the WDM network only transmits on 1 wavelength, then no two calls can share the same link. We say that colouring $\gamma = \{(r_1, \lambda_1), \ldots, (r_n, \lambda_n)\}$ is a *valid colouring for 1 wavelength* iff no vertices r_i and r_j ($i \neq j$) are connected by an edge in G. In general we have the following

Definition 7 (Valid colouring) *We say $\gamma = \{(r_1, \lambda_1), \ldots, (r_n, \lambda_n)\}$ is a valid colouring iff no vertices $r_1 \neq r_2$ connected in G exist such that $\{(r_1, \lambda), (r_2, \lambda)\} \subseteq \gamma$ for some λ.*

We make the following

Definition 8 (Constraint satisfaction) *We say that a routing ρ k-satisfies a constraint set $\mathcal{C} \subseteq \mathcal{P}(R) \times \mathbb{N}$ iff*

$$\forall (l, c) \in \mathcal{C} : \left(\sum_{(r,i) \in \rho_l} i \right) \leq c \cdot k,$$

where $\rho_l = \{(r, i) \in \rho \mid r \in l\}$.

A subset of the nodes of G is said to be *independent* iff no two nodes are connected by an edge of G. An independent set is *maximum* if no other independent sets with more elements exist. The independence number $\alpha(H)$ of a graph H is the size of a maximum independent set in H. The set of nodes in a subgraph H of G with $\alpha(H) = c$ is called a *c-independent set*. A c-independent set is *maximal* if it is not a proper subset of a larger c-independent set.

For instance, $\{BC, CD\}$ of Figure 5.2 on the preceding page is a maximal 1-independent set, and all the nodes of the interference graph constitute a maximal 2-independent set. Note that while Ramaswami and Sivarajan (1995) consider maximal independent node sets in G for *directly* computing an optimal routing, we consider maximal c-independent *subgraphs* H of G for $c = 1, \ldots, \alpha(G)$ with an aim to extract interference information that *indirectly* describes what optimal routings are possible.

We now define

$$\mathcal{C}_0 = \{(l, c) \mid l \text{ is a maximal } \ c\text{-independent set}\}$$

and can then show

Lemma 9 (Satisfaction implication) *A routing ρ k-satisfies $\mathcal{C}_0$ if there exists a valid colouring γ for k wavelengths such that* $\text{routes}(\gamma) = \rho$.

Proof: Let a valid colouring γ be given with $\text{routes}(\gamma) = \rho$. Assume ρ does not k-satisfy $\mathcal{C}_0$, i.e. for some $(l, c) \in \mathcal{C}_0$ we have $\sum_{(r,i) \in \rho_l} i > c \cdot k$. Restricting γ to l we obtain a valid k'-colouring γ_l where $k' \leq k$, in which each colour $1, \ldots, k'$

is used at most $\alpha(l)$ times. This makes a total of at most $\alpha(l) \cdot k' \leq c \cdot k$ times, contradicting the assumption. □

Informally, this key lemma states that a necessary condition for being able to assign free wavelengths to a routing $\rho = \{(r_1, i_1), \ldots, (r_n, i_n)\}$ in a network with k wavelengths on each link is that ρ k-satisfies $\mathcal{C}_0$.

For $k = 2$ wavelengths we are able to prove the reverse implication. First we let $\mathcal{C}_0' = \{(l', c') \mid l' \text{ is a simple loop } \wedge c' = \max(1, \lfloor |l'|/2 \rfloor)\}$. A simple loop in the interference graph is a cyclic, connected subgraph where each node has arity 2, i.e. it does not "cross itself". It is obvious that its independence number is $\max(1, \lfloor |l'|/2 \rfloor)$, so for $(l', c') \in \mathcal{C}_0'$, l' is a c-independent set, and by the definition of maximality there exists $(l, c) \in \mathcal{C}$ such that $l' \subseteq l$ and $c' = c$. This implies that $\rho_{l'} \subseteq \rho_l$ in Definition 8 on the preceding page, so if ρ 2-satisfies $\mathcal{C}_0$, we find for any $(l', c') \in \mathcal{C}_0'$ that $\left(\sum_{(r,i)\in\rho_{l'}} i\right) \leq \left(\sum_{(r,i)\in\rho_l} i\right) \leq c \cdot k$; in other words, that ρ also 2-satisfies $\mathcal{C}_0'$.

Thus what remains to be shown is

Lemma 10 *If a routing ρ 2-satisfies $\mathcal{C}_0'$, then there exists a valid colouring γ for 2 wavelengths such that* $\text{routes}(\gamma) = \rho$.

Proof: Let ρ be given and assume that all colourings γ with $\text{routes}(\gamma) = \rho$ are invalid.

Now define a *state* as $(M, \gamma) \in \mathcal{P}(R) \times \mathcal{P}(R \times \Lambda)$ and construct a state transformer that operates on these states. We ensure that in every transformation step $(M, \gamma) \to (M', \gamma')$,

- M' only contains coloured vertices,
- γ' specifies a valid colouring of M' for 2 wavelengths
- $\text{routes}(\gamma) = \text{routes}(\gamma')$
- M' can be written as a union of disjoint sets $M' = M_1 \cup \cdots \cup M_k$, where each M_i is a connected graph, and at most one M_i has a node with a coloured neighbour $r' \notin M_i$[1].

The initial state is set to (M_0, γ_0), where γ_0 is some (invalid) colouring with $\text{routes}(\gamma_0) = \rho$, and $M_0 = \{r_0\}$ for some coloured vertex r_0. The state transformer considers the possible cases:

[1]This means that the distance between any two M_i, M_j, $i \neq j$ is at least 2 links—they do not "touch" each other.

Case 1: If for all $r \in M$, r has no coloured neighbour $r' \notin M$, M is "isolated" from any remaining coloured nodes by a "wall" of uncoloured nodes, and we consider

Case 1.1: If there are no coloured vertices $r' \notin M$, then γ is a valid colouring for M and thus for the whole of R, which contradicts the assumption.

Case 1.2: If there exists a coloured vertex $r' \notin M$, add r to M.

Case 2: If there exists $r \in M$ that has a coloured neighbour $r' \notin M$, we consider

Case 2.1: If for all $r \in M$, r has no neighbour with the same colour as r, select some neighbour $r' \notin M$ and add r' to M.

Case 2.2: If there exists $r \in M$ that has a neighbour $r' \notin M$ with the same colour as r, we consider

Case 2.2.1: If r or r' is coloured with two colours, the loop $l = \{r, r'\}$ has at least 3 colours and does not satisfy the constraint $(l, 1) \in \mathcal{C}_0'$.

Case 2.2.2: If r and r' each have just one colour, we consider

Case 2.2.2.1: If r' has no neighbour $r'' \in M \setminus \{r\}$ with the opposite colour of r', then change the colour of r' (i.e. change γ) and add r' to M.

Case 2.2.2.2: If r' has a neighbour $r'' \in M \setminus \{r\}$ with the opposite colour of r', then there must be some loop $l = l' \cup \{r'\}$ where $l' \in M$ (cf. Figure 5.3), where the length of l is $2n+3$ for some $n \in \mathbb{N}$, and all vertices in l are coloured. Thus ρ does not 2-satisfy the constraint $(l, n+1) \in \mathcal{C}_0'$.

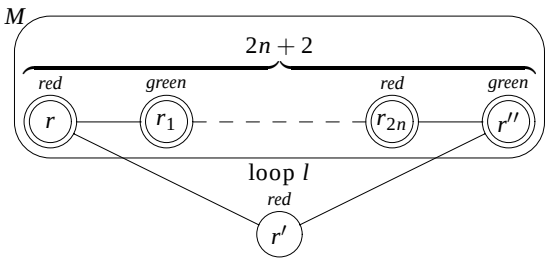


Figure 5.3: If r' interferes, $(l, n) \in \mathcal{C}_0'$ is not satisfied

In every transformation step, the size of $R \setminus M$ is never decreased, and either we reach a contradiction, we find that ρ does not 2-satisfy $\mathcal{C}_0'$, or the size of $R \setminus M$ is decreased by 1. As the latter cannot go on forever (R is assumed finite), we conclude that ρ does not 2-satisfy $\mathcal{C}_0'$. □

Proving this lemma for $k = 1$ is even simpler: Assume routes(γ) does not 1-satisfy $\mathcal{C}_0'$. Then there exists $(l,c) \in \mathcal{C}_0'$ such that $\sum_{(r,i)\in\rho_l} i > c$, i.e. more than $\max(1, \lfloor |l|/2 \rfloor)$ routes in loop l are coloured, which means that there are two coloured vertices r_1 and r_2 connected by an edge in G.

5.2.2 Minimising constraint sets

We observe that given vertex sets $l \subseteq l_1 \cup \cdots \cup l_n$ with capacities $c \geq c_1 + \cdots + c_n$, if the number of coloured vertices in $l_1 \cup \cdots \cup l_n$ is no greater than $c_1 + \cdots + c_n$, the same is true for l. This leads to

Definition 11 (Constraint reduction) *Given a set of constraints $\mathcal{C}$, redundant constraints are eliminated by the function*

$$\text{reduce}(\mathcal{C}) = \left\{ (l,c) \in \mathcal{C} \;\middle|\; \begin{array}{c} \forall\, \mathcal{C}' \subseteq \mathcal{C} \setminus \{(l,c)\} : \\ l \subseteq \bigcup\limits_{(l',c')\in\mathcal{C}'} l' \Rightarrow c < \sum\limits_{(l',c')\in\mathcal{C}'} c' \end{array} \right\}$$

Given the constraint set shown in Table 5.3, the result of reducing it is shown in Table 5.4 on the following page. For instance, the constraint $(\{AB, BC, CD, DE, BE\}, 2)$ is eliminated by constraints $(\{AB, BC, BE\}, 1)$ and $(\{CD, DE\}, 1)$.

sets l	c
$\{AB\}, \{AE\}, \{BC\}, \{BE\}, \{CD\}, \{DE\}$	1
$\{AB, AE\}, \{AB, BC\}, \{BC, CD\}, \{CD, DE\}, \{DE, AE\}, \{BE, DE\}, \{AE, BE\}, \{AB, BE\}, \{BC, BE\}$	1
$\{AB, BE, AE\}, \{AB, BC, BE\}, \{AE, BE, DE\}$	1
$\{AB, BC, BE, AE\}, \{AB, AE, DE, BE\}, \{BC, CD, DE, BE\}$	2
$\{AB, BC, CD, DE, AE\}, \{AB, BC, CD, DE, BE\}, \{AE, BE, BC, CD, DE\}, \{AB, BC, BE, DE, AE\},$	2
$\{AB, BC, CD, DE, BE, AE\}$	2

Table 5.3: A set of constraints

This reduction is correct in the sense that it preserves k-satisfaction; to prove this, we first need to show

sets l	c
$\{BC, CD\}, \{CD, DE\}, \{AB, BE, AE\}, \{AB, BC, BE\}, \{AE, BE, DE\}$	1
$\{AB, BC, CD, DE, BE, AE\}$	2

Table 5.4: Reduced constraint set from Table 5.3

Lemma 12 *Define*

$$p(\mathcal{C}', l, c) = \left(l \subseteq \sum_{(l',c') \in \mathcal{C}'} l' \wedge c \geq \sum_{(l',c') \in \mathcal{C}'} c' \right).$$

If $(l, c) \in \mathcal{C} \setminus \text{reduce}(\mathcal{C})$ *then there exists a* $\mathcal{C}' \subseteq \text{reduce}(\mathcal{C})$ *such that* $p(\mathcal{C}', l, c)$ *holds.*

Proof: By definition of reduce there exists a $\mathcal{C}_i \subseteq \mathcal{C} \setminus \{(l, c)\}$ such that $p(\mathcal{C}_i, l, c)$. If $\mathcal{C}_i \subseteq \text{reduce}(\mathcal{C})$ we are done. Otherwise, suppose that $(l_1, c_1) \in \mathcal{C}_i \setminus \text{reduce}(\mathcal{C})$. Then there exists a $\mathcal{C}_i' \subseteq \mathcal{C} \setminus \{(l_1, c_1)\}$ such that $p(\mathcal{C}_i', l_1, c_1)$. This implies that $p(\mathcal{C}_{i+1}, l, c)$, where $\mathcal{C}_{i+1} = \mathcal{C}_i \setminus \{(l_1, c_1)\} \cup \mathcal{C}_i'$. The process can now be repeated, each iteration removing an element (l_1, c_1) not in $\text{reduce}(\mathcal{C})$.

Defining the partial lexicographical ordering $\sqsubseteq \subseteq (\mathcal{P}(R) \times \mathbb{N}) \times (\mathcal{P}(R) \times \mathbb{N})$ by $(l_2, c_2) \sqsubseteq (l_1, c_1) = c_2 < c_1 \vee (c_2 = c_1 \wedge l_2 \supseteq l_1)$, we see that in each iteration, for any element (l_2, c_2) introduced into $\mathcal{C}_{i+1}$ (by $\mathcal{C}_i'$) we have $(l_2, c_2) \sqsubset (l_1, c_1)$. As $\sqsubseteq$ is a well-founded ordering, the process will terminate with a final iteration I, and the lemma will hold for $\mathcal{C}' = \mathcal{C}_{I+1}$. □

Lemma 13 (Reduction correctness) *Given a routing* ρ *and a set of constraints* $\mathcal{C}$, ρ *k-satisfies* $\text{reduce}(\mathcal{C})$, *iff* ρ *k-satisfies* $\mathcal{C}$.

Proof: **"if":** Obvious, because $\text{reduce}(\mathcal{C}) \subseteq \mathcal{C}$.

"only if": Pick $(l, c) \in \mathcal{C}$. If $(l, c) \in \text{reduce}(\mathcal{C})$, we see that $\sum_{(r,i) \in \rho_l} i \leq c \cdot k$ is trivial, so let us now assume $(l, c) \notin \text{reduce}(\mathcal{C})$. This implies by Lemma 12 that there exists a constraint set $\mathcal{C}' \subseteq \text{reduce}(\mathcal{C})$ such that

$$l \subseteq \bigcup_{(l',c') \in \mathcal{C}'} l' \wedge c \geq \sum_{(l',c') \in \mathcal{C}'} c'.$$

So if we define $\rho_l' = \{(r, i) \in \rho \mid r \in \bigcup_{(l',c') \in \mathcal{C}'} l'\}$, we have $\rho_l \subseteq \rho_l'$. As $\mathcal{C}' \subseteq \text{reduce}(\mathcal{C})$, we have by assumption

$$\sum_{(r,i) \in \rho_{l'}} i \leq c' \cdot k \text{ for all } (l', c') \in \mathcal{C}',$$

so

$$\sum_{(l',c')\in\mathcal{C}'} \sum_{(r,i)\in\rho_{l'}} i \leq \sum_{(l',c')\in\mathcal{C}'} c' \cdot k \leq c \cdot k.$$

Finally, we find

$$\sum_{(r,i)\in\rho_l} i \leq \sum_{(r,i)\in\rho'_l} i \leq \sum_{(l',c')\in\mathcal{C}'} \sum_{(r,i)\in\rho_{l'}} i \leq c \cdot k.$$

□

5.2.3 Multifibre extension

Until now we have only considered networks in which each link holds just one fibre with k wavelengths. We now consider networks where links can have any number of fibres, each fibre still containing k wavelengths. The connections are still circuit-switched, but the continuity constraint is slightly relaxed for links with several fibres: Given a path and a wavelength, the connection can be made if for each link the wavelength is unused on at least one fibre. For instance, given the network shown in Figure 5.4, up to two connections can be made along route

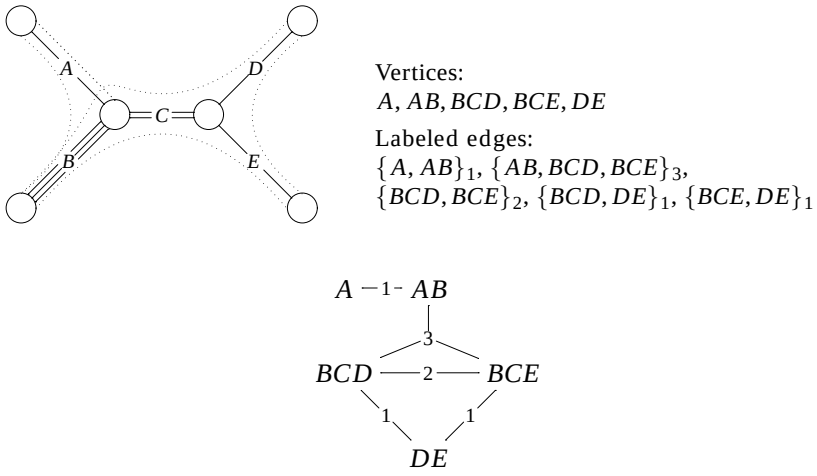


Figure 5.4: WDM network with multifibre links, and its interference graph with generalised edges

BCD and *BCE* on the same wavelength without being blocked, because link *C*

contains two fibres. Similarly, up to three connections can be made along routes AB, BCD and BCE, because link B contains three fibres.

We treat this multifibre case by generalising the interference graph to an edge-labelled graph, as can be seen in the figure. An edge is now generalised to be a subset of vertices (i.e. routes), and each edge label is an integer indicating the minimum number of fibres on any link on which the routes interfere.

We let a *marked* set of vertices be a set of vertices where each is associated with a positive integer. We define a marked subset of vertices in an edge-labelled graph G to be *independent*, if for each edge connecting vertices of this subset, the sum of the marks on the vertices, minus the edge label, is positive. The previous definition in Section 5.2.1 is a special case of this, where all edge labels and vertex marks were equal to 1. An independent marked set is *maximum* if no other independent set with a higher mark sum exists, and $\alpha(H)$ is the sum of marks in a maximum independent marked set in H.

Finally, we generate constraints as before. For the WDM network which is shown in Figure 5.4 on the page before, some constraints are

$(\{A, AB\}, 1)$, $(\{BCD, DE\}, 1)$, $(\{BCE, DE\}, 1)$, $(\{BCD, BCE, DE\}, 2)$, $(\{A, AB, BCD, BCE, DE\}, 3)$.

and removing the superfluous constraints, we are left with

$(\{A, AB\}, 1)$, $(\{BCD, DE\}, 1)$ and $(\{BCE, DE\}, 1)$.

5.3 Algorithms

The algorithm we propose is shown schematically in Figure 5.5 on the facing page. Given a WDM network in link/route representation (cf. Table 5.1 on page 176 and Definition 4 on page 177), we construct the interference graph. Conceptually, we then find all c-independent sets (including the maximal ones) for $c = 1, \ldots$, which includes the constraint set $\mathcal{C}_0$, and remove superfluous constraints by Definition 11 on page 181 (cf. Tables 5.3 and 5.4). In practice, constraint reduction is performed each time a constraint is added to the set, to keep the constraint set size small. The reduced constraint set is then fed into an existing blocking calculation algorithm, leading to the end-to-end blocking probabilities for the routes of the WDM network.

What we calculate is in fact an approximation of the end-to-end blocking. However, if we assume dynamic rerouting and unbounded node switching capacity, we obtain the exact blocking probability when using $k = 1$ and $k = 2$ colours, cf. Lemma 10 on page 179.

Finally we note that the constraint set reduction does not change these properties, as seen by Lemma 13 on page 182.

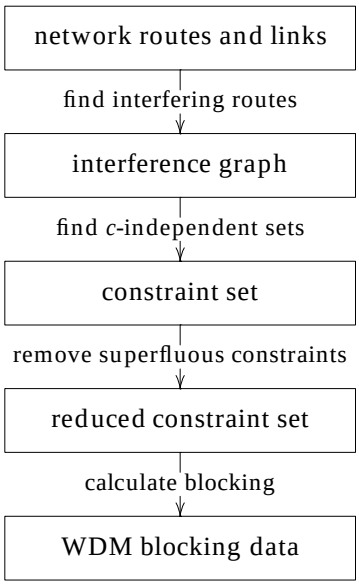


Figure 5.5: Overall WDM blocking calculation

5.3.1 Calculating constraints

Given the interference graph G, we calculate the c-independent sets, and thus the constraints, using the basic, recursive algorithm shown in Figure 5.6. The al-

```
FindMCIsets(R) =
  excludenodes ← {}
  for r ∈ R do
    startnode ← {r}
    MCISets(startnode, excludenodes, {})
    excludenodes ← excludenodes ∪ {r}

MCISets(caninclude, mustexclude, cset) =
  if caninclude ≠ {} then
    r ← pop(caninclude)
    newcset ← cset ∪ {r}
    newinclude ← (caninclude ∪ neighbours(r))
                 \ (newcset ∪ mustexclude)
    /* Consider subgraphs containing r */
    MCISets(newinclude, mustexclude, newcset)
    /* Consider subgraphs not containing r */
    MCISets(caninclude \ {r}, mustexclude ∪ {r}, cset)
  else
    c ← α(cset)
    if not IsSuperfluous((cset, c), C0) then
      C0 ← C0 ∪ {(cset, c)}
```

Figure 5.6: Finding maximal c-independent sets, storing them in $\mathcal{C}_0$

gorithm enumerates all possible connected subgraphs of the interference graph, and combines each with its independence number $\alpha(cset)$ to form a constraint. If the constraint is not superfluous with respect to the constraints already in $\mathcal{C}_0$, it is added. The function α for calculating the independence number is shown in Figure 5.8 on the next page; cf. Figure 5.9 on page 188 for the multifibre extension. As the function is recursive, many sets have their independence number

```
IsSuperfluous((l, c), C) =
  if l = {} ∧ c ≥ 0 then return true      /* ({}, c) is always satisfied for c ≥ 0   */
  if c ≤ 0 then return false              /* (l, c) is not superfluous for c ≤ 0     */
  if C = {} then return false             /* Singletons are not superfluous          */
  C' ← {(l', c') ∈ C | l' ∩ l ≠ {} ∧ c' ≤ c}   /* (l', c') could make (l, c) superfluous */
  sort C' ascendingly according to c'/|l' ∩ l|   /* Consider strongest constraints first */
  to obtain C' = {(l_1, c_1), ..., (l_n, c_n)}
  for i ∈ {1, ..., n}
    C'' ← {(l_{i+1}, c_{i+1}), ..., (l_n, c_n)}
    if IsSuperfluous((l \ l_i, c − c_i), C'') then   /* Subtract (l_i, c_i) and call recursively */
      return true
  return false
```

Figure 5.7: Detecting superfluous constraints

```
α(rs) =
  a ← 0
  if rs ≠ {} then
    r ← pop(rs)
    rs' ← rs \ {r}
    a ← α(rs')
    rs' ← rs' \ neighbours(r)
    a ← max(a, 1 + α(rs'))
  return a
```

Figure 5.8: Calculating the independence number

$\alpha(rs, c) =$
  $a \leftarrow 0$
  if $rs \neq \{\}$ **then**
    $r \leftarrow \mathrm{pop}(rs)$
    $rs' \leftarrow rs \setminus \{r\}$
    $a \leftarrow \alpha(rs', c)$
    $b \leftarrow 0$
    repeat
      $b \leftarrow b + 1$
      for $e \in \{e \in E \mid r \in e\}$ **do** $c(e) \leftarrow c(e) - 1$
      $rs' \leftarrow rs' \setminus \{r' \in \mathrm{neighbours}(r) \mid \forall e \in E :$
                $\{r, r'\} \subseteq e \Rightarrow c(e) \leq 0\}$
      $a \leftarrow \max(a, b + \alpha(rs', c))$
    until $\forall e \in E : r \in e \Rightarrow c(e) \leq 0$
  return a

Figure 5.9: Calculating the independence number in edge-labeled graphs

recalculated many times, so the results are cached for better performance.

5.3.2 Removing constraints

Superfluous constraints are detected using the recursive algorithm shown in Figure 5.7 on page 187. By iterating through all the constraints in $\mathcal{C}_0$, checking for each one whether it is superfluous with respect to the remaining constraints, we reduce the constraint set to contain only strictly necessary constraints.

5.4 Empirical studies

In this section we report some results from empirical studies of the differences in blocking between WDM and ES (electrically switched) networks (or, equivalently, WDM networks with wavelength conversion everywhere). We use the algorithm described in the preceding sections to convert ES network specifications (link / route tables) into WDM network specifications (constraint / route tables). Both kinds of tables are then used as input to an algorithm devised by Pinsky and Conway (1992) to calculate the exact end-to-end blockings on the routes, using the Blocked Calls Lost model with Poisson arrivals.

Two kinds of networks, simple and "real-world" networks, are studied for several kinds of route patterns, as shown in Figures 5.10–5.11 and 5.12:

Simple star networks with 5, 6 and 7 links using

- all-to-all routes
- asymmetric route pattern
- slightly asymmetric route pattern

Simple ring networks with 5, 6 and 7 links using

- all-to-all routes along a shortest path
- all routes of length 2

Simple double ring networks with 5, 6 and 7 links using

- all-to-all routes along a shortest path
- all routes of length 1 & 2, and for the 7 link case also length 3

A fictive Danish national network with 11 links

Three kinds of traffic loads, illustrated in Figure 5.13 on page 193, are considered for the simple networks:

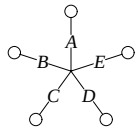
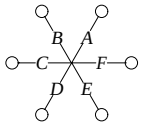
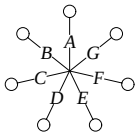
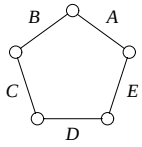
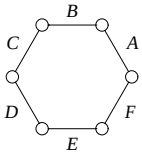
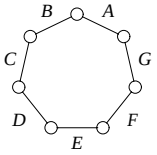
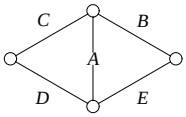
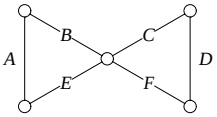
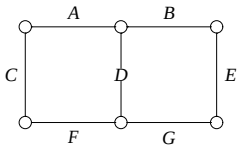
Links	5	6	7
Star			
Ring			
Double ring			

Figure 5.10: Simple 5, 6 and 7 link star, ring and double ring networks

- *Link balanced traffic,* where we attempt to make the offered load on each link be 1 erlang
- *Path uniform traffic,* where all paths have the same offered load, and we attempt to make the load on most links be close to 1 erlang
- *Length dependent traffic,* where the offered load is inversely proportional to the length of the path, and we attempt to make the load on most links be close to 1 erlang.

The traffic loads for the Danish national network are computed according to a gravitational model where the offered traffic between two cities is proportional to the product of the population of the two regions serviced by the cities, and normalised (divided) by a factor to make the offered load on the most loaded link equal to 1 erlang.

Some examples of the computed interference graphs and WDM constraint / route tables are shown in Figure 5.14 on page 194; results for all the networks and traffic patterns are shown in Appendix B. In general, the constraint / route tables for the WDM networks have more constraints than their ESN (electrically switched network) counterparts, but usually only a few more; the Dan-

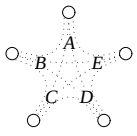
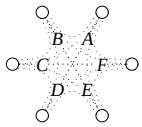
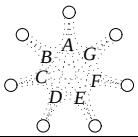
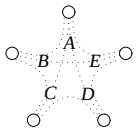
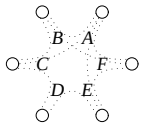
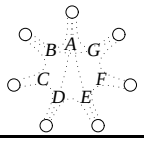
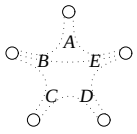
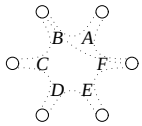
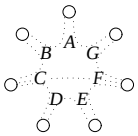
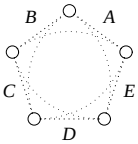
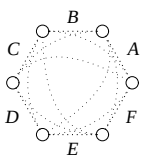
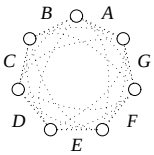
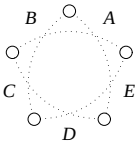
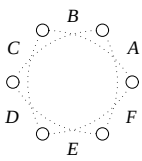
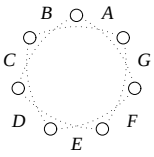
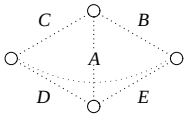
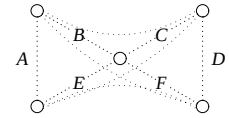
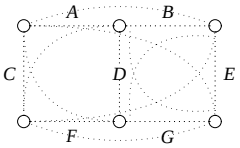
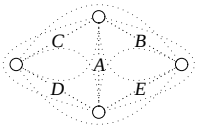
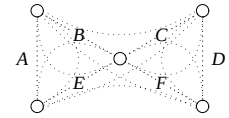
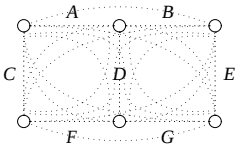
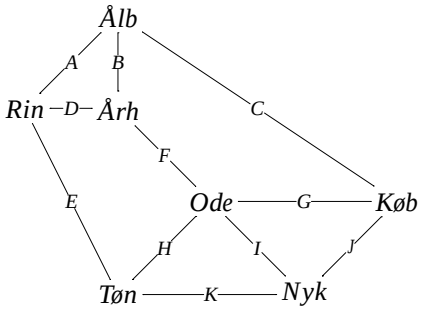
Links	5	6	7
A star: all-to-all routes			
B star: asymmetric route pattern			
C star: slightly asymmetric route pattern			
A ring: all-to-all routes along a shortest path			
B ring: routes of length 2			
A double ring: all-to-all routes along a shortest path			
B double ring: routes of length 1 & 2 (&3 for 7 links)			

Figure 5.11: Simple 5, 6 and 7 link star, ring and double ring network routes



	Ålb	Rin	Årh	Ode	Køb	Tøn	Nyk
Ålb		4	24	16	28	8	12
Rin	A		6	4	7	2	3
Årh	B	D		24	42	12	18
Ode	BF	DF	F		28	8	12
Køb	C	DFG	FG	G		14	21
Tøn	AE	E	FH	H	JK		6
Nyk	CJ	EK	FI	I	J	K	

Figure 5.12: Danish national network with 11 links, fewest hops routing and gravitational traffic

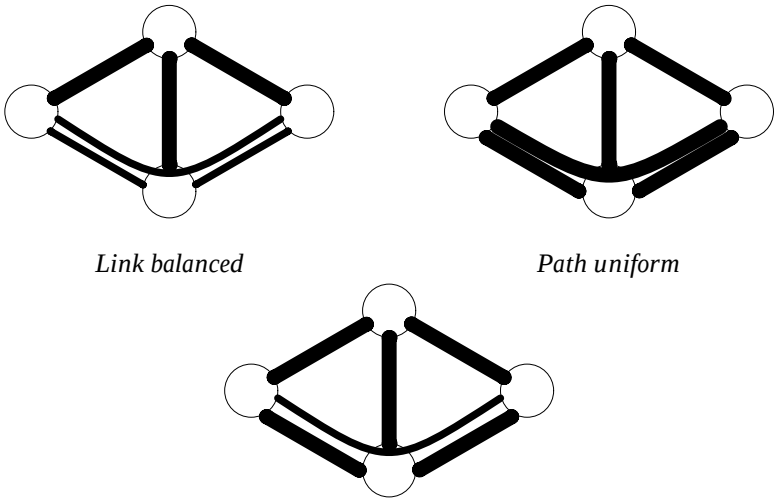


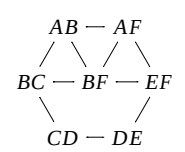
Figure 5.13: Three different traffic pattern types

ish network, for instance, includes just one extra constraint in the WDM table: $CG + CJ + EH + EJK + GH \leq 2 \cdot k$. The largest WDM table is for the 7 link C star WDM network, which requires 64 constraints. For the A double ring networks and the 6 link B ring network, the ESN and WDM tables, and thus the blocking, are identical.

We try to obtain results for $n = 1, \ldots, 8$ connections per link; for large WDM constraint / route tables and high values of n, the blocking calculation algorithm does not terminate within reasonable time, though. For an experiment with n connections, the offered traffic is multiplied by n, and the capacities c for the links / constraints of the ES / WDM networks are also multiplied by n. Due to statistical multiplexing, the blocking is thus lower for higher values of n.

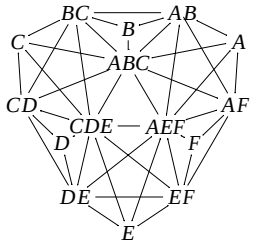
As well as the absolute blockings for the ES and WDM networks, we also calculate the relative blocking increase from the ES to the WDM network. Complete results are given in Appendix B; Figures 5.15–5.18 show the results for the 6 link C star network, using the following offered traffic (length dependent traffic is in this case identical to path uniform traffic):

	AB	*BC*	*CD*	*DE*	*AE*	*BE*
Link balanced	0.500	0.250	0.750	0.250	0.500	0.250
Path uniform	0.250	0.250	0.250	0.250	0.250	0.250



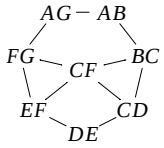
6 link C star

AB	AF	BC	BF	CD	DE	EF	c
1		1	1				1
		1		1			1
				1	1		1
					1	1	1
	1		1			1	1
1	1		1				1
		1	1	1	1	1	2



6 link A ring

AB	BC	CD	DE	EF	FG	AG	CF	c
1						1		1
1	1							1
	1	1					1	1
		1	1					1
			1	1				1
				1	1		1	1
					1	1		1
1	1	1	1	1	1	1	1	3
1	1				1	1	1	2



7 link C star

Figure 5.14: Examples of interference graphs and constraint / route tables

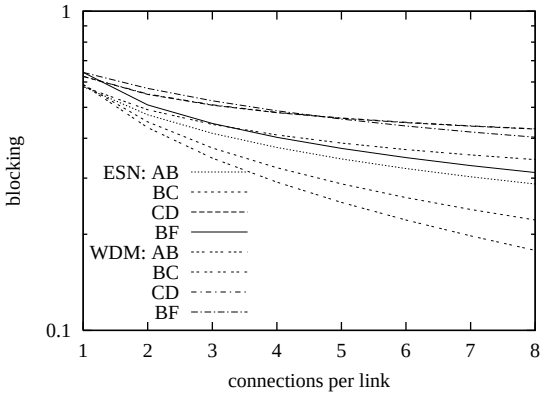


Figure 5.15: Blocking of link balanced traffic in the 6 link C star network

As has been shown in previous work (Glenstrup and Iversen, 2001; Kovačević and Acampora, 1995), some routes in the WDM network (e.g., route *BC*) experience *lower* blocking than their ESN counterparts, at the expense of others experiencing higher blocking. Contrary to what one might expect, the individual ESN blocking values are thus *not* lower bounds of the corresponding WDM blocking values—however, the total carried traffic in the network is lower.

We also note that the relative blocking difference in the star network gets worse for higher values of n, cf. Figures 5.16 and 5.18.

Typically, though, the relative blocking differences between ES and WDM networks get smaller for higher values of n. As an example, link balanced traffic for the 6 link A ring network, where the offered traffic is 0.2 erlang for all routes except *B*, *D* and *F* which are loaded with 0.4 erlang, results in the blocking shown in Figure 5.19 on page 197 and blocking increase shown in Figure 5.20 on page 198. Moreover, in most networks, the relative blocking difference is very small: typically less than 0.08 for the double ring and star networks (except the 5 link B and C stars, and the 6 link C star, which experience up to 30% higher blocking), and less than 0.04 for the ring networks. We also note from Figure 5.20 on page 198 a general tendency: shorter paths experience a lower relative blocking increase than longer paths—which confirms previous results in the literature (Barry and Humblet, 1996; Barry and Marquis, 1995; Chlamtac et al., 1992). The reason for this is obvious: The shorter the path, the higher the chance of finding a common free wavelength, and for 1-link routes the wavelength is insignificant. The differ-

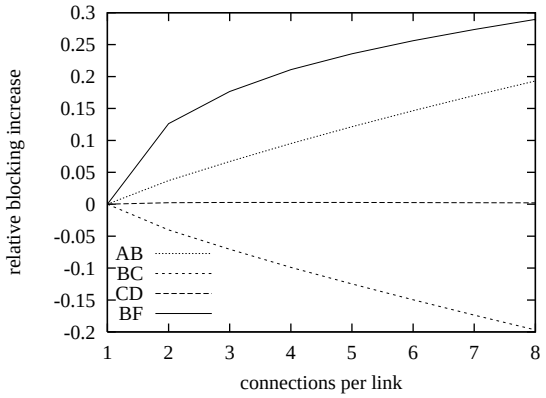


Figure 5.16: Blocking increase of link balanced traffic from ESN to WDM in the 6 link C star network

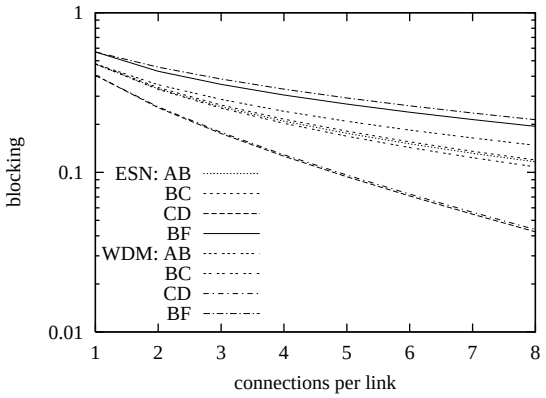


Figure 5.17: Blocking of path uniform traffic in the 6 link C star network

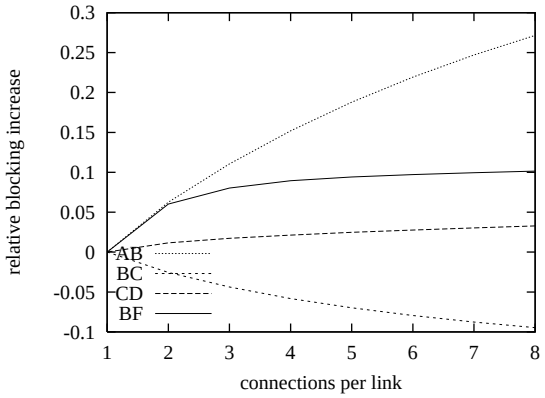


Figure 5.18: Blocking increase of path uniform traffic from ESN to WDM in the 6 link C star network

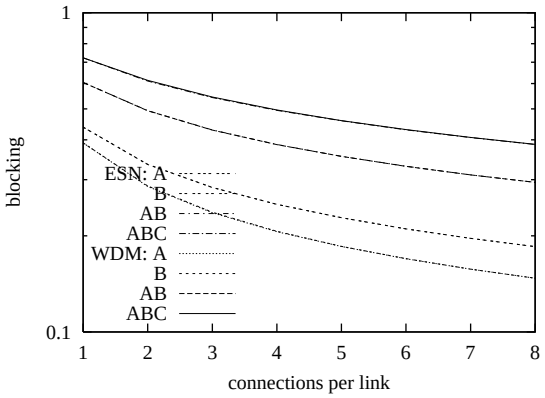


Figure 5.19: Blocking of link balanced traffic in the 6 link A ring network

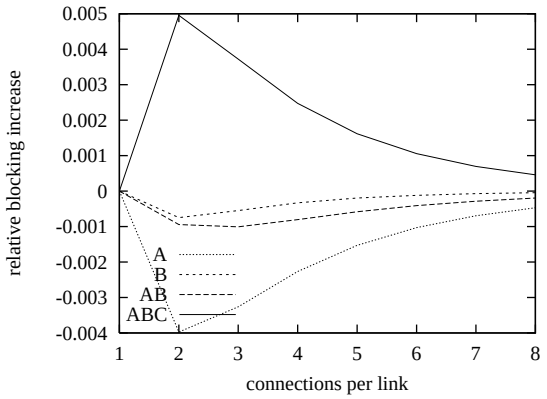


Figure 5.20: Blocking increase of link balanced traffic from ESN to WDM in the 6 link A ring network

ent load patterns (link balanced / path uniform / length dependent) do not change the overall shapes of the relative blocking graphs, except that in general, higher loaded paths experience a higher *relative* blocking increase in WDM networks, compared to their ESN counterparts.

For the Danish national network, the differences in blocking between the ESN and WDM network are very small; some results are shown in Figures 5.21–5.23.

5.5 Static routing in WDM networks

The method presented here is also useful when assessing routing of static traffic demands. Consider the 5-link A-star routes shown in Figure 5.11 on page 191 and their constraint / route Table 5.5 on page 200. The top part above the dotted line is the link / route table of the ESN. Assume we must route uniform traffic, that is, one wavelength per route. Adding up the digits of each of the top five lines we see that the ESN would require $k = 4$ lines on each link. However, we see from the constraint immediately below the dotted line ($2 \cdot k \geq 1+1+1+1+1+1+1+1+1+1$) that $k = 4$ is insufficient for the WDM network and that $k = 5$ wavelengths per fibre are required to route the static traffic.

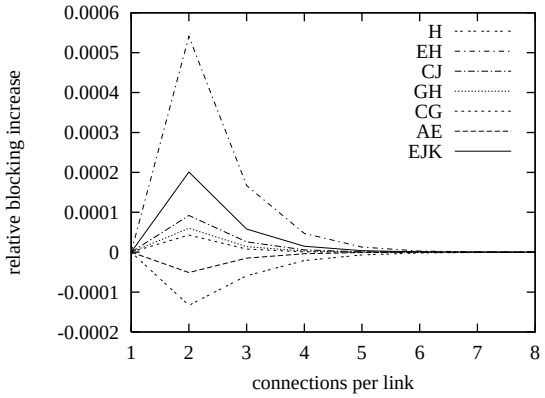


Figure 5.21: Blocking increase from ESN to WDM in the Danish network, I

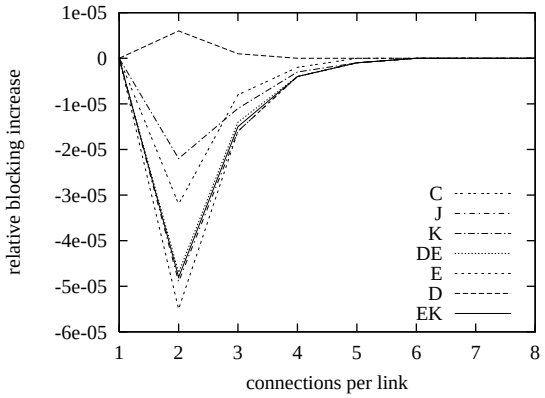


Figure 5.22: Blocking increase from ESN to WDM in the Danish network, II

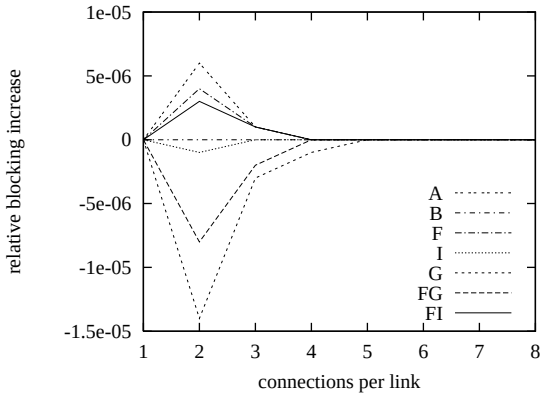


Figure 5.23: Blocking increase from ESN to WDM in the Danish network, III

AB	*AC*	*AD*	*AE*	*BC*	*BD*	*BE*	*CD*	*CE*	*DE*	*c*
1	1	1	1	0	0	0	0	0	0	1
1	0	0	0	1	1	1	0	0	0	1
0	1	0	0	1	0	0	1	1	0	1
0	0	1	0	0	1	0	1	0	1	1
0	0	0	1	0	0	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	2
1	1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	1	0	0	0	1
0	1	1	0	0	0	0	1	0	0	1
0	1	0	1	0	0	0	0	1	0	1
0	0	1	1	0	0	0	0	0	1	1
0	0	0	0	1	1	0	1	0	0	1
0	0	0	0	1	0	1	0	1	0	1
0	0	0	0	0	1	1	0	0	1	1
0	0	0	0	0	0	0	1	1	1	1

Table 5.5: Constraint / route table for the 5-link A star routes

5.6 Conclusions

We have formally defined the wavelength routing and colouring concepts in WDM networks, and, based on these firm definitions, we have shown how a WDM link / route table can be converted to a general constraint / route table which can be used as input to blocking calculation algorithms. The blocking values thus calculated approximate the WDM blocking values, and thus enable us to re-use all the existing blocking calculation tools for electrically circuit-switched networks.

We have implemented an algorithm that performs this transformation, and have applied it in a systematic empirical study to networks with 5, 6 and 7 links, as well as a Danish national network.

The results show that in general some paths in the WDM network experience higher blocking, and some lower blocking, than their counterparts in the ES network. The difference in blocking is, however, typically very small, and only some special cases experience a relative blocking increase of 30%. In general, shorter paths experience a lower relative increase in blocking than longer paths, and in some cases more heavily loaded paths risk higher blocking increases than more lightly loaded paths.

Part IV

Conclusion

Chapter 6

Conclusion

In this thesis we developed several software tools for solving greenfield network design problems, as well as tools for analysing the blocking probabilities in circuit-switched WDM networks without wavelength converters.

In Chapter 2 we focused specifically on optimising network design considering link costs composed of duct costs and traffic-dependent fibre costs. We developed a two-phased tool in which the first phase generates a set of promising paths from which the second phase, a heuristic based on simulated allocation, can select paths to reduce the network deployment cost. Evaluations showed that the promising path generator supplied path sets of good quality for the nominal network design task but that the path sets for protected network design were not quite as good. The heuristic was compared with ILP optimisation tools and with a commercially available network design tool and turned out to produce fairly good results.

The design tool was extended in Section 3.3 to include the nodes as an integrated part of the optimisation, using a second degree polynomial for modelling node costs. Experiments showed that the tool was able to reduce overall node costs with up to 20% in some examples for the nominal design task, while the tool is not able to achieve significant savings for the protection design.

Section 3.1 was devoted to node costs, introducing a multigranular optical switch model that reduces the size of the switch matrix by grouping and switching wavelengths in bands and fibres. Our novel algorithm, subpath wavelength grouping (SWG), is able to optimise a multigranular network for both link and node costs by routing in an unorthodox way: starting with common subpaths instead of individual connections. Empirical evaluations showed that compared to previous methods SWG is effective at reducing network costs, especially when link costs are relatively higher than node port costs.

In Section 3.2 we considered the problem of subdividing wavelengths into timeslots to match the data transport granularity with the traffic demand granularity. We showed that designing a wavelength and timeslot routed optical network is equivalent to designing just a wavelength routed optical network with more wavelengths, and further that wavelength or timeslot conversion and timeslot delay had no effect on the wavelength usage in the network. Furthermore, we showed that optimised shortest path routing is good at minimising wavelength usage when using a hop-distance metric.

In Section 3.4 we analysed the popular MS-SPRing networks, constructing an ILP model as an arc-flow model augmented with constraints modelling the rings. We showed that a real world ring network designed by a human network planner could be designed at a cheaper cost using by running the model through an ILP solver; the general lesson learnt was to consider using a lot of rings of various sizes, rather than using few rings with ring interworking. Moreover, we found by comparing real world prices that mesh network switch prices must be reduced by approximately 40% if they are to compete on cost with the MS-SPRing networks.

Chapter 4 reported on preliminary experiments with an MPλS connection setup simulator that also models node and link delays. The results seemed to indicate that node delays have a more severe effect on call blocking than link delays, and furthermore that a single wavelength reservation strategy generally produces the lowest blocking probabilities.

Finally, in Chapter 5 we presented a novel way of calculating analytically the blocking probabilities in a WDM network by extending the link / route table with additional constraints representing the wavelength continuity constraint. The constraints were computed by transforming the original WDM network problem into a graph colouring problem and extracting maximal c-independent sets from the graphs. We extended the basic model to include multifibre links, and we made some empirical studies which showed that in shorter paths generally experience lower blocking when wavelength converters are not used, while longer paths experience higher blocking—however, the difference in blocking is typically insignificant.

We conclude that the goal set up in the introduction of this thesis has been met, and we have fulfilled the objective of contributing to computer assisted network planning. Naturally, no matter how much ground any research covers, new and intriguing unanswered questions will remain, and we shall list a few of the more interesting ones in the following section.

6.1 Future work

- One of the main problems seen during experiments with the greenfield network design optimiser described in Chapter 2 and Section 3.3 is that the paths generated by the PPG algorithm in the first phase set the limits for how good a solution the heuristic in the second phase can produce, especially for protected network design. It would be interesting to iterate the two phases, letting feedback from the optimisation phase support the generation of better paths—this would especially be useful when considering integrated node and link optimisation for the protection design scenario.

- The SWG algorithm which is developed in Section 3.1 assumes the existence of wavelength converters at every port, but it would be useful to extend the algorithm to a scenario of no converters, so that the difference in network resource usage and the significance of wavelength conversion in multigranular networks can be assessed.

- Another interesting extension would be to model parametric wavelength converters (Antoniades et al., 1999; Escobar and Marshall, 2002) that allow concurrent but limited wavelength conversion of all wavelengths in one band.

- For MS-SPRing networks to be fully protected against single failures special ring interworking schemes must be used to protect traffic traversing several rings. It would be interesting to see whether the ILP model in Section 3.4 could be extended to include for example “drop & continue” ring interworking.

- The MPλS simulator described in Chapter 4 suffers from the same problem as all other simulators, namely that it takes very long time to reach accurate results for traffic sources of low intensity. Developing an analytic model of the MPλS setup delay scenario, even an approximate one, could alleviate this problem.

- However, the analytical calculations performed in Chapter 5 are of high complexity and thus not applicable to real world networks of any significant size. A way of circumventing this problem could be to combine them with iterative decomposition methods in a style similar to that used by Zhu et al. (2000).

Bibliography

Emile Aarts and Jan Korst. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley, December 1998.

Alok Aggarwal, Amotz Bar-Noy, Don Coppersmith, Rajiv Ramaswami, Baruch Schieber, and Madhu Sudan. Efficient routing in optical networks. *Journal of the ACM*, 46(6): 973–1001, November 1996.

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.

AixCom. COMPOSIS®, 2002. URL `http://www.aixcom.com`.

Neophytos Antoniades, S.J. Ben Yoo, Krishna Bala, Georgios Ellinas, and Thomas E. Stern. An architecture for a wavelength-interchanging cross-connect utilizing parametric wavelength converters. *Journal of Lightwave Technology*, 17(7):1113–1125, 1999.

Grenville Armitage. MPLS: The magic behind the myths. *IEEE Communications Magazine*, pages 124–131, January 2000.

Daniel Awduche and Yakov Rekhter. Multiprotocol lambda switching: Combining MPLS traffic engineering control with optical crossconnects. *IEEE Communications Magazine*, pages 111–116, March 2001.

Ayan Banerjee, John Drake, Jonathan P. Lang, Brad Turner, Daniel Awduche, Lou Berger, Kireeti Kompella, and Yakov Rekhter. Generalized multiprotocol label switching: An overview of signaling enhancements and recovery techniques. *IEEE Communications Magazine*, 39(7):144–151, July 2001a.

Ayan Banerjee, John Drake, Jonathan P. Lang, Brad Turner, Kireeti Kompella, and Yakov Rekhter. Generalized multiprotocol label switching: An overview of routing and management enhancements. *IEEE Communications Magazine*, 39(1):144–150, January 2001b.

Stefano Baroni and Polina Bayvel. Wavelength requirements in arbitrarily connected wavelength-routed optical networks. *Journal of Lightwave Technology*, 15(2):242–251, 1997.

Stefano Baroni, Polina Bayvel, Richard J. Gibbens, and Steven K. Korotky. Analysis and design of resilient multifiber wavelength-routed optical transport networks. *Journal of Lightwave Technology*, 17(5):743–758, May 1999.

Richard A. Barry and Pierre A. Humblet. Models of blocking probability in all-optical networks with and without wavelength changers. *IEEE Journal on Selected Areas in Communications*, 14(5):858–867, June 1996.

Richard A. Barry and Douglas Marquis. An improved model of blocking probability in all-optical networks. In *Digest of LEOS Summer Topical Meetings*, pages 43–44, 1995.

Pietro Belotti and Thomas Jacob Kjær Stidsen. Optimal placement of wavelength converting nodes. In *Third International Workshop on Design of Reliable Communication Networks*, December 2001.

A. Bianco, S. Binetti, P.N. Caponio, A. Hill, E. Leonardi, M. Listanti, F. Neri, and R. Sabella. Dimensioning of a single layer optical platform based on the 'switchless' concept for large scale networks. In *Proceedings of the IEEE Lasers and Electro-Optics Society Annual Meeting (LEOS '99)*, volume 2, pages 521–522. IEEE, 1999.

S. Bigo, S. Gauchard, S. Borne, P. Bousselet, P. Poignant, L. Lorcy, A. Bertaina, J.-P. Thiery, S. Lanne, L. Pierre, D. Bayart, L.-A. de Montmarillon, R. Sauvageon, J.-F. Chariot, P. Nouchi, J.-P. Hamaide, and J.-L. Beylat. 1.5 terabit/s WDM transmission of 150 channels at 10 Gbit/s over 4*100 km of TeraLight/sup TM/fibre. In *Proceedings of the 25th European Conference on Optical Communication (ECOC 1999)*, pages 40–1, 1999.

Alexander Birman. Computing approximate blocking probabilities for a class of all-optical networks. *IEEE Journal on Selected Areas in Communications*, 14(5):852–857, June 1996.

D. Blair. Impact of dial-up internet access traffic on PSTN/ISDN dimensioning. In *Proceedings of 14th European Network Planning Workshop (ENPW '98)*, page paper no. 1.3, Les Arcs, March 1998.

Anthony Brooke, David Kendrick, and Meeraus Alexander. *GAMS, A Users Guide*. Boyd & Fraser Publishing Company, 1992.

Maurici Garcia Brustenga. Teletraffic problems in WDM networks. Master's thesis, Technical University of Denmark, Lyngby, Denmark, June 1999.

Michel W. Chbat, Emmanuel Grard, Luc Berthelon, Amaury Jourdan, Philippe A. Perrier, Alain Leclert, B. Landousies, A. Ramdane, Noel Parnis, Edwin V. Jones, Emmanuel Limal, Henrik N. Poulsen, R.J.S. Pedersen, N. Flaaronning, Danny Vercauteren, Mario Puelo, Ernesto Ciaramella, Giuseppe Marone, R. Hess, H. Melchior, Wim Van Parys, Piet M. Demeester, P. J. Gødsvang, Torodd Olsen, and Dag R. Hjelme. Toward wide-scale all-optical transparent networking: The ACTS optical pan-european network (OPEN) project. *IEEE Journal on Selected Areas in Communications*, 16(7), September 1998.

Imrich Chlamtac, Aura Ganz, and Gadi Karmi. Purely optical networks for terabit communication. In *Proceedings of IEEE INFOCOM 1998*, volume 3, pages 887–896. IEEE Computing Society Press, 1989.

Imrich Chlamtac, Aura Ganz, and Gadi Karmi. Lightpath communications: An approach to high bandwidth optical WAN's. *IEEE/ACM Transactions on Networking*, 40(7):1171–1182, 1992.

Thomas Kaare Christensen, Bo Friis Nielsen, and Villy Bæk Iversen. Distribution of channel holding times in cellular communication systems. In Johan M. Karlsson, Ulf Körner, and Christian Nyberg, editors, *Fifteenth Nordic Teletraffic Seminar (NTS–15)*, pages 185–193. Lund Institute of Technology, Lund, Sweden, August 2000.

Ernesto Ciaramella. Introducing wavelength granularity to reduce the complexity of optical cross connects. *IEEE Photonics Technology Letters*, 12(6):699–701, 2000.

Tibor Cinkler. ILP formulation of grooming over wavelength routing with protection. In A. Jukan, editor, *Towards an Optical Internet. New Visions in Optical Network Design and Modelling. IFIP TC6 Fifth Working Conference on Optical Network Design and Modelling (ONDM 2001)*, pages 25–47. Kluwer Academic, 2002.

Tibor Cinkler, Dániel Marx, Claus Popp Larsen, and Dániel Fogaras. Heuristic algorithms for joint configuration of the optical and electrical layer in multi-hop wavelength routing networks. *Proceedings of IEEE INFOCOM 2000*, 2:1000–1009, March 2000.

William J. Cook, William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver. *Combinatorial Optimization*. Wiley, 1998.

Steven Cosares, David Deutsch, Iraj Saniee, and Ondria Wasem. Sonet toolkit: A decision support system for designing robust and cost-effective fiber-optic networks. *Interfaces*, 25, 1995.

Sebastian Cwilich, Mei Deng, David F. Lynch, S.J. Phillips, and J.R. Westbrook. Algorithms for restoration planning in a telecommunications network. *Lecture Notes in Computer Science*, 1619:194–209, 1999.

Jose Augusto de Azevedo, Joaquim Joao E.R. Silvestre Madeira, Ernesto Q. Vieira Martins, and Filipe Manual A. Pires. Computational improvement for a shortest paths ranking algorithm. *European Journal of Operational Research*, 73(1):188–191, 1994.

Adam Dickmeiss and Morten Larsen. Spærringsberegninger i telenet. Master's thesis, Department of Telecommunication, Technical University of Denmark, 1993.

Bharat Doshi, Subrahmanyam Dravida, P. Harshavardhana, Oded Hauser, and Yfei Wang. Optical network design and restoration. *Bell Labs Technical Journal*, January–March 1999.

Emmanuel Dotaro, Dimitri Papadimitriou, Ludovic Noirie, Laurant Ciavaglia, and Martin Vigoureux. Optical multi-granularity architectural framework. Technical report, IPO Working Group Internet Draft, July 2001. draft-dotaro-ipo-multi-granularity-00.txt.

Rudra Dutta and George N. Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. Technical Report TR-99-06, Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534, May 1999.

Jaafar M.H. Elmirghani and Hussein T. Mouftah. All-optical wavelength conversion: Technologies and applications in DWDM networks. *IEEE Communications Magazine*, pages 86–92, March 2000.

Hector E. Escobar and Larry R. Marshall. All-optical wavelength band conversion enables new scalable and efficient optical network architectures. In *Proceedings of the Conference on Optical Fiber Communication (OFC 2002)*, pages WH2: 225–227, 2002.

Johannes Färber, Stefan Bodamer, and Joachim Charzinski. Statistical evaluation and modelling of internet dial-up traffic. In *Proceedings of the SPIE—The International Society for Optical Engineering*, volume 3841, pages 112–121. Society of Photo-Optical Instrumentation Engineers, 1999.

Christian Fenger and Arne John Glenstrup. Synchronous optical hierarchy. In *Proceedings of the International Seminar on Telecommunication Networks and Teletraffic Theory*, St. Petersburg, January–February 2002.

Christian Fenger, Emmanuel Limal, Ulrik Gliese, and Cathal J. Mahon. Statistical study of the correlation between topology and wavelength usage in optical networks with and without conversion. In *Networking 2000*, volume 1815 of *Lecture Notes in Computer Science*, pages 168–175, Paris, France, May 2000. Springer-Verlag.

Albert A. Fredericks. Impact of holding time distributions on parcel blocking in multi–class networks with application to internet traffic on PSTN's. In Peter Key and David Smith, editors, *Proceedings of the International Teletraffic Congress (ITC–16)*, volume 3b, pages 877–886. Elsevier Science, June 1999.

Piero Gambini, Monique Renaud, Christian Guillemot, Franco Callegati, Ivan Andonovic, Bruno Bostica, Dominique Chiaroni, Giorgio Corazza, Søren Lykke Danielsen, Philippe Gravey, Peter Bukhave Hansen, Michel Henry, Christopher Janz, Allan Kloch, Roger Krähenbühl, Carla Raffaelli, Michael Schilling, Anne Talneau, and Libero Zucchelli. Transparent optical packet switching: Network architecture and demonstrators in the KEOPS project. *IEEE Journal on Selected Areas in Communications*, 16(7), September 1998.

Linda Morales Gardner, M. Heydari, J. Shah, I. Hal Sudborough, Ioannis G. Tollis, and C. Xia. Techniques for finding ring covers in survivable networks. *Proceedings of the Global Telecommunications Conference (GLOBECOM '94)*, 3:1862–1866, 1994.

Linda Morales Gardner, I. Hal Sudborough, and Ioannis G. Tollis. NetSolver: A software tool for the design of survivable networks. In *IEEE Global Telecommunication Conference*, volume 2, pages 926–930. IEEE, IEEE, 1995.

Michael R. Garey and David S. Johnson. *Computers and Intractability. A guide to the theory of NP-completeness.* Freeman, 1979.

Mathieu Garnot and F. Masetti. Dimensioning and design of the WDM optical layer in transport networks. In *Proceedings of the SPIE—The International Society for Optical Engineering*, volume 3230, pages 244–252. Society of Photo-Optical Instrumentation Engineers, 1997.

Mathieu Garnot and Phillipe A. Perrier. Planning of WDM networks: Methods, routing node modeling and applications. In *Proceedings of the Global Telecommunications Conference (GLOBECOM '98)*, volume 1, pages 351–355. IEEE, November 1998.

Mathieu Garnot, M. Sotom, and F. Masetti. Routing strategies for optical paths in WDM networks. In *Proceedings of the IEEE International Conference on Communications (ICC–97)*, volume 1, pages 422–426. IEEE, 1997.

Ori Gerstel, Rajiv Ramaswami, and Weyl-Kuo Wang. Making use of a two stage multiplexing scheme in a WDM network. In T. Li, editor, *Proceedings of the Conference on Optical Fiber Communication (OFC 2000)*, volume 3, pages 44–46. Optical Society of America, March 2000.

Arne John Glenstrup, Christian Fenger, and Thomas Jacob Kjær Stidsen. Full design of robust optical networks. In Johan M. Karlsson, Ulf Körner, and Christian Nyberg, editors, *Fifteenth Nordic Teletraffic Seminar (NTS–15)*. Lund Institute of Technology, Lund University, August 2000.

Arne John Glenstrup and Villy Bæk Iversen. Exact evaluation of blocking in WDM networks. In *Mini-conference on Teletraffic*, pages 3.7: 121–126, St. Petersburg, June 2001.

Arne John Glenstrup, Naohide Nagatsu, Jun Yamawaku, and Wataru Imajuku. Impact of new wavelength grouping technique based on the subpath concept. In *Proceedings of the IEICE Society Conference*, pages B–10–100, 2001.

Wayne D. Grover and Demetrios Stamatelakis. Bridging the ring-mesh dichotomy with *p*-cycles. In *Proceedings of the 2nd International Workshop on Design of Reliable Communcation Networks (DRCN 2000)*, pages 92–104, Munich, Germany, April 2000. Technical University of Munich, Herbert Utz Verlag.

Christian Guillemot, Monique Renaud, Piero Gambini, Christopher Janz, Ivan Andonovic, Raimond Bauknecht, Bruno Bostica, Marco Burzio, Franco Callegati, Maurizio Casoni, Dominique Chiaroni, Fabrice Clerot, Søren Lykke Danielsen, Francios Dorgeuille, Arnaud Dupas, A. Franzen, Peter Bukhave Hansen, David K. Hunter, Allan Kloch, R. Krähenbühl, Bruno Lavigne, Alain Le Corre, Carla Raffaelli, Michael Schilling, Jean-Claude Simon, and Libero Zucchelli. Transparent optical packet switching: The european ACTS KEOPS project approach. *Journal of Lightwave Technology*, 16(12):2117–2134, 1998.

Kazunari Harada, Kenji Shimizu, Teruhiko Kudou, and Takeshi Ozeki. Hierarchical optical path cross-connect systems for large scale WDM networks. In *Proceedings of the Conference on Optical Fiber Communication (OFC 1999)*, volume 2, pages 356–358. IEEE, February 1999.

D.R. Hjelme and S.E. Andersen. Designing physical topologies for wavelength-routed all-optical networks. In G. de Marchis and R. Sabella, editors, *Optical Networks: Design and Modelling. IFIP TC6 Second Working Conference on Optical Network Design and Modelling (ONDM '98)*, pages 167–174. Kluwer Academic, 1999.

Hong Huang and J. Copeland. Hamiltonian cycle protection: A novel approach to mesh WDM optical network protection. In *IEEE Workshop on High Performance Switching and Routing*, pages 31–35. IEEE, 2001.

Nen-Fu Huang, Guan-Hsiung Liaw, and Chuan-Pwu Wang. A novel all-optical transport network with time-shared wavelength channels. *IEEE Journal on Selected Areas in Communications*, 18(10):1863–1875, October 2000.

David K. Hunter and Ivan Andonovic. Approaches to optical internet packet switching. *IEEE Communications Magazine*, 38(9):116–122, September 2000.

David K. Hunter, Meow C. Chia, and Ivan Andonovic. Buffering in optical packet switches. *Journal of Lightwave Technology*, 16(12):2081–2094, 1998a.

David K. Hunter, W. David Cornwell, Tim H. Gilfedder, Andre Franzen, and Ivan Andonovic. SLOB: a switch with large optical buffers for packet switching. *Journal of Lightwave Technology*, 16(10):1725–1736, 1998b.

David K. Hunter, M.H.M. Nizam, Meow C. Chia, Ivan Andonovic, K.M. Guild, A. Tzanakaki, M.J. O'Mahony, L.D. Bainbridge, M.F.C. Stephens, R.V. Penty, and I.H. White. WASPNET: a wavelength switched packet network. *IEEE Communications Magazine*, 37 (3):120–129, 1999.

Ibrar Hussain. Characterization of dial-up based internet user traffic. Master's thesis, Department of Telecommunication, Technical University of Denmark, August 1999. 89pp.

ILOG Cooperation. *ILOG CPLEX 6.5, Users Manual*, March 1999.

ILOG Cooperation. *ILOG CPLEX 7.0, Users Manual*, August 2000.

Villy Bæk Iversen. Analyses of real teletraffic processes based on computerized measurements. *Ericsson Technics*, 29(1):3–64, 1973.

Villy Bæk Iversen. The exact evaluation of multi-service loss system with access control. In *Teleteknik, English Ed.*, volume 31, pages 56–61, 1987.

Villy Bæk Iversen. *Introduction to Traffic Engineering*. Department of Telecommunication, Technical University of Denmark, 1998. 460pp.

Villy Bæk Iversen. *Data- og Teletrafikteori*. Den Private Ingeniørfond, Technical University of Denmark, January 1999.

Villy Bæk Iversen, Arne John Glenstrup, and Jens Rasmussen. Internet dial-up traffic modelling. In Johan M. Karlsson, Ulf Körner, and Christian Nyberg, editors, *Fifteenth Nordic Teletraffic Seminar (NTS–15)*, pages 67–78. Lund Institute of Technology, Lund University, August 2000.

Rong-Hong Jan, Fung-Jen Hwang, and Sheng-Tzong Cheng. Topological optimization of a communication network subject to a reliability constraint. *IEEE Transactions on Reliability*, 42(1):63–70, March 1993.

Neil D. Jones. *Computability and Complexity from a Programming Perspective*. Foundations of Computing. MIT Press, Boston, London, 1 edition, 1997.

Rajgopal Kannan, Radim Bartos, Kyungsook Y. Lee, and Harry F. Jordan. STWnet: A high bandwidth space-time-wavelength multiplexed optical switching network. In *Proceedings of IEEE INFOCOM 1997*, volume 2, pages 777–784. IEEE, 1997.

Aaron Kershenbaum, Parviz Kermani, and George A. Grover. MENTOR: An algorithm for mesh network topological optimization and routing. *IEEE Transactions on Communications*, 39(4):503–513, April 1991.

E. Scott Kirkpatrick, C. Daniel Gelatt, Jr., and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

Milan Kovačević and Anthony Acampora. On wavelength translation in all-optical networks. In *Proceedings of IEEE INFOCOM 1995*, volume 2, pages 413–422. IEEE, 1995.

Jeremy Lawrence. Designing multiprotocol label switching networks. *IEEE Communications Magazine*, 39(7):134–142, July 2001.

Chun-Ming Lee, Chi-Chun R. Hui, Frank F.-K. Tong, and Peter T.-S. Yum. Network dimensioning in WDM-based all-optical networks. In *Proceedings of the Global Telecommunications Conference (GLOBECOM '98)*, volume 1, pages 328–333. IEEE, 1998a.

Kyungsik Lee, Kyungchul Park, Sungsoo Park, and Heesang Lee. Economic spare capacity planning for DCS mesh-restorable networks. *European Journal of Operational Research*, 110(1):63–75, 1998b.

Myungmoon Lee, Jintae Yu, Yongbum Kim, Kang Chul-Hee, and Jinwoo Park. Design of hierarchical crossconnect WDM networks employing a two-stage multiplexing scheme of waveband and wavelength. *IEEE Journal on Selected Areas in Communications*, 20(1): 166–171, 2002.

Taehan Lee, Kyungsik Lee, and Sungsoo Park. Optimal routing and wavelength assignment in WDM ring networks. *IEEE Journal on Selected Areas in Communications*, 18(10): 2146–2154, October 2000.

Christopher Leith, Glen Takahara, and Zouheir Mansourati. Exact, approximate and recursive blocking in all-optical networks. In *Proceedings of the Global Telecommunications Conference (GLOBECOM '99)*, pages 1460–1465, 1999.

Lightscape Networks. LightPlan^{TM}, 2002. URL `http://www.lightscapenetworks.com`.

Rao Lingampalli and Praveen Vengalam. Effect of wavelength and waveband grooming on all-optical networks with single layer photonic switching. In *Proceedings of the Conference on Optical Fiber Communication (OFC 2002)*, pages ThP4: 501–502, 2002.

Marco Listanti and Vincenzo Eramo. Architectural and technological issues for future optical internet networks. *IEEE Communications Magazine*, pages 82–92, September 2000.

Henrik Listov-Saabye and Villy Bæk Iversen. "ATMOS," a PC-based tool for evaluating multi-service telephone systems. Technical report, IMSOR, Technical University of Denmark, 1989. 75pp (in Danish).

Hannan Luss, Moshe B. Rosenwein, and Richard T. Wong. Topological network design for sonet ring architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(6): 780–790, November 1998.

G. Mohan, C. Siva Ram Murthy, and Arun K. Somani. Efficient algorithms for routing dependable connections in WDM optical networks. *IEEE/ACM Transactions on Networking*, 9(5):553–566, October 2001.

Anne Mette Møller and Jens Rasmussen. Impact of internet traffic on the telephone network. In *ENPW'98*, pages 15–21, Les Arcs, France, 1998.

G. Dave Morley and Wayne D. Grover. Current approaches in the design of ring-based optical networks. In M. Meng, editor, *Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, volume 1, pages 220–225. IEEE, May 1999.

Naohide Nagatsu, Satoru Okamoto, and Ken-ichi Sato. Optical path cross-connect system scale evaluation using path accommodation design for restricted wavelength multiplexing. *IEEE Journal on Selected Areas in Communications*, 14(5):893–902, June 1996.

M. Naldi. Internet access through the telephone network: An analysis of traffic patterns. In *Proceedings of 14th European Network Planning Workshop (ENPW '98)*, page Paper no. 1.1, Les Arcs, March 1998.

M. Naldi. Measurement-based modelling of internet dial-up access connections. *Computer Networks*, 31(22):2381–2390, 1999.

Ludovic Noirie, C. Blaizot, and Emmanuel Dotaro. Multi-granularity optical cross-connect. In *Proceedings of the 26 th European Conference on Optical Communication (ECOC 2000)*, volume 3, pages 269–270, September 2000.

Ludovic Noirie, Martin Vigoureux, and Emmanuel Dotaro. Impact of intermediate traffic grouping on the dimensioning of multi-granularity optical networks. In *Proceedings of the Conference on Optical Fiber Communication (OFC 2001)*, volume 2, pages TuG3–1–TuG3–3. IEEE, March 2001.

OPNET Technologies, Inc. OPNET modeler®, 2002. URL `http://www.opnet.com`.

Conny Palm. Intensity variations in telephone traffic. *Ericsson Technics*, 44:189pp, 1987. Translated from "Intensitätsschwankungen im Fernsprechverkehr" (1943).

Harry G. Perros. Computer simulation techniques: The definitive introduction! North Carolina State University, 1999.

Eugene Pinsky and Adrian E. Conway. Computational algorithms for blocking probabilities in circuit-switched networks. *Annals of Operations Research*, 35:31–41, 1992.

Michał Pióro. Solving multicommodity integral flow problems by simulated allocation. *Telecommunication Systems*, 7(1–3):17–28, 1997.

Michał Pióro. ATM network design. Technical Report P10352–1, Department of Communication Systems, Lund Institute of Technology, Lund University, Lund, Sweden, 1999. SWAP Project.

Michał Pióro, Thomas Jacob Kjær Stidsen, Arne John Glenstrup, Christian Fenger, and Henrik Christiansen. Design problems in robust optical networks. In *Networks 2000*, September 2000.

Rajiv Ramaswami and Kumar N. Sivarajan. Routing and wavelength assignment in all-optical networks. *IEEE/ACM Transactions on Networking*, 3(5):489–500, 1995.

Rajiv Ramaswami and Kumar N. Sivarajan. *Optical Networks: A practical perspective*. Systems on Silicon. Morgan Kaufmann Publishers Inc., San Fransisco, California, 1998.

R. Ryf, J. Kim, J.P. Hickey, A. Gnauck, D. Carr, F. Pardo, C. Bolle, R. Frahm, N. Basavanhally, C. Yoh, D. Ramsey, R. Boie, R. George, J. Kraus, C. Lichtenwalner, R. Papazian, J. Gates, H.R. Shea, A. Gasparyan, V. Muratov, J.E. Griffith, J.A. Prybyla, S. Goyal, C.D. White, M.T. Lin, R. Ruel, C. Nijander, S. Arney, D.T. Neilson, D.J. Bishop, P. Kolodner, S. Pau, C. Nuzman, A. Weis, B. Kumar, D. Lieuwen, V. Aksyuk, D.S. Greywall, T.C. Lee,

H.T. Soh, W.M. Mansfield, S. Jin, W.Y. Lai, H.A. Huggins, D.L. Barr, R.A. Cirelli, G.R. Bogart, K. Teffeau, R. Vella, H. Mavoori, A. Ramirez, N.A. Ciampa, F.P. Klemens, M.D. Morris, T. Boone, J.Q. Liu, J.M. Rosamilia, and C.R. Giles. 1296-port MEMS transparent optical crossconnect with 2.07 petabit/s switch capacity. In *Proceedings of the Conference on Optical Fiber Communication (OFC 2001)*, volume 4, pages PD28–1–3, 2001.

Debashis Saha. Forward reservation protocol with immediate unlock (FRP-IU) for dynamic lightpath establishment in all-optical networks (AONs). In *Proceedings of the SPIE—The International Society for Optical Engineering*, volume 4213, pages 234–241, 2000.

Iraj Saniee. Optimal routing designs in self-healing communication networks. *International Transactions in Operational Research*, 3(2):187–195, 1996.

Ken-ichi Sato. *Advances in Transport Network Technologies: Photonic Networks, ATM, and SDH*. Artech House, Boston • London, 1996.

Sudipta Sengputa and Ramu Ramamurthy. From network design to dynamic provisioning and restoration in optical cross-connect mesh-networks: An architectural and algorithmic overview. *IEEE Network*, pages 46–54, July–August 2001.

Mike Sexton and Andy Reid. *Broadband Networking: ATM, SDH, and SONET*. Artech House, Boston • London, 1997.

Sajjad H. Shami and Mark C. Sinclair. Co-evolutionary agents for telecommunication network restoration. *Proceedings of Recent Advances in Soft Computing '99*, pages 146–151, 1999.

J.M. Simmons and A.A.M. Saleh. The value of optical bypass in reducing router size in gigabit networks. In *Proceedings of the IEEE International Conference on Communications (ICC–99)*, volume 1, pages 591–596, June 1999.

Mark C. Sinclair. Optical mesh network topology design using node-pair encoding genetic programming. *Proceedings of Genetic and Evolutionary Computation Conference (GECCO-99)*, pages 191–199, July 1999.

Jan Späth and Stefan Bodamer. Performance evaluation of photonic networks under dynamic traffic conditions. In G. de Marchis and R. Sabella, editors, *Optical Networks: Design and Modelling. IFIP TC6 Second Working Conference on Optical Network Design and Modelling (ONDM '98)*, pages 13–21. Kluwer Academic, 1999.

Ashwin Sridharan and Kumar N. Sivarajan. Blocking in all-optical networks. In *Proceedings of IEEE INFOCOM 2000*, volume 2, pages 990–999, March 2000.

Aravind Srinivasan. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. In *Proceedings of 38th Annual Symposium on Foundations of Computer Science*, pages 416–425. IEEE Computing Society Press, October 1997.

Demetrios Stamatelakis and Wayne D. Grover. Theoretical underpinnings for the efficiency of restorable networks using preconfigured cycles ("*p*-cycles"). *IEEE Transactions on Communications*, 48(8):1262–1265, August 2000.

Thomas Jacob Kjær Stidsen and Arne John Glenstrup. Quantifying optimal mesh and ring design costs. *Naval Research Logistics*, page 14pp, 2002. Submitted.

Yong Sun, Jun Gu, and Danny H.K. Tsang. Routing and wavelength assignment in all-optical networks with multihop connections. *International Journal of Electronics and Communications*, 55(1):8, 2001.

J.W. Suurballe. Disjoint paths in a network. *Networks*, 4(2):125–45, 1974.

Tellium. StarNet Planner, 2002. URL `http://www.tellium.com`.

Tushar Tripathi and Kumar N. Sivarajan. Computing approximate blocking probabilities in wavelength routed all-optical networks with limited-range wavelength conversion. *IEEE Journal on Selected Areas in Communications*, 18(10):2123–2129, October 2000.

Griselda Navarro Varela and Mark C. Sinclair. Ant colony optimisation for virtual-wavelength-path routing and wavelength allocation. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC '99)*, volume 3, pages 1809–1816. IEEE, 1999.

Malathi Veeraraghavan, Ramesh Karri, Tim Moors, Mark Karol, and Reinette Grobler. Architectures and protocols that enable new applications on optical networks. *IEEE Communications Magazine*, 39(3):118–127, March 2001.

VPIsystems^{TM}. VPItransportMaker, 2002. URL `http://www.vpisystems.com`.

A. Watanabe, Satoru Okamoto, M. Koga, Ken-ichi Sato, and M. Okuno. Design and performance of delivery and coupling switch board for large scale optical path cross-connect system. *IEICE Transactions on Communications*, E81-B(6):1203–12, 1998.

S. Westerberg. The distribution of the number of scans between successive calls when scanning a poisson call arrival process. *Ericsson Technics*, 31(1):37–53, 1975.

Gordon Wilfong, Benny Mikkelsen, Chris Doerr, and Martin Zirngibl. WDM cross-connect architectures with reduced complexity. *Journal of Lightwave Technology*, 17(10): 1732–1741, 1999.

Lunchakorn Wuttisittikulkij, Charoenchai Baworntummarat, and Thanyaporn Iamvasant. A comparative study of mesh and multi-ring designs for survivable WDM networks. *IEICE Transactions on Communications*, E83-B(10):2270–2277, 2000.

Xipeng Xiao and Lionel M. Ni. Internet QoS: A big picture. *IEEE Network*, 13(2):8–18, March / April 1999.

Jun Yamawaku, Wataru Imajuku, Atsushi Takada, Satoru Okamoto, and Morioka Toshio. Investigation of virtual grouped-wavelength-path routing networks. In *Proceedings of the International Conference on Integrated Optics and Optical Fiber Communication, and the Optoelectronics and Communication Conference (IOOC/OECC '01)*, pages 194–196, Sydney, Australia, July 2001.

Jin Y. Yen. Finding the *k* shortest loopless paths in a network. *Management Science*, 17: 712–716, 1971.

Hui Zang, Jason P. Jue, and Biswanath Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Optical Networks Magazine*, 1(1):47–60, January 2000.

Zhensheng Zhang and Anthony S. Acampora. A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength re-use. *IEEE/ACM Transactions on Networking*, 3(3):281–288, 1995.

Yuhong Zhu, George N. Rouskas, and Harry G. Perros. A path decomposition approach for computing blocking probabilities in wavelength-routing networks. In *IEEE/ACM Transactions on Networking*, volume 8, pages 747–762. IEEE, 2000.

List of abbreviations

Abbreviation	*Abbreviated concept*	*Page*
ADSL	Asymmetrical digital subscriber line	274
AON	All-optical network	6
ATM	Asynchronous transfer mode	11
BPP	Binomial–Poisson–Pascal	288
BXC	Band cross-connect	74
DXC	Digital cross-connect	6
E2EG	End-to-end grouping	72
ES	Electrically switched	189
ESN	Electrically switched network	189
FDL	Fibre-optic delay lines	93
FXC	Fibre cross-connect	74
GMPLS	Generalised multiprotocol label switching	12
ILP	Integer linear programming	15
IP	Internet protocol	11
IR	Integrated routing	133
ISDN	Integrated services digital network	274
LP	Linear programming	8
LSP	Label switched path	12
MPLS	Multi-protocol label switching	12
MPλS	Multi-protocol wavelength switching	12
MS-SPRing	Multiplex section shared protection ring	9
MST	Minimum spanning tree	58
NGMD	Normalised general mean difference	288
ND	Nominal design	27
OADM	Optical add / drop multiplexer	6
OEO	Optical-electrical-optical	6

Abbreviation	*Abbreviated concept*	*Page*
OXC	Optical cross-connect	6
PDP	Path diversity protection	27
PPG	Promising path generator	25
PR	Prerouting	133
PSTN	Public switched telephony system	274
RAM	Random access memory	64
RWA	Routing and wavelength assignment	7
SAL	Simulated allocation	24
SAN	Simulated annealing	24
SOH	Synchronous optical hierarchy	90
SBP	Single backup path	27
SBP+NRPC	Single backup path with non-reusable primary capacity	27
SBP+FW	Single backup path with fixed wavelength	27
SWG	Subpath wavelength grouping	70
TRP	Total rerouting protection	27
TSP	Travelling salesperson	58
WDM	Wavelength division multiplexed	6
WRON	Wavelength routed optical network	7
WXC	Wavelength cross-connect	74

Index

Part V

Appendices

Appendix A

Algorithms for subpath wavelength grouping

This appendix describes in more detail the algorithms used in Section 3.1 to realise subpath wavelength grouping.

A.1 Switch ports

In Step 4 of the algorithm on page 77, a *port* p is a description of fibre numbers, band numbers and wavelength numbers—this can be used to describe where a group of wavelengths are switched to or from:

Port	$\ni$	p	::=	$[(f_1, l_1), \ldots, (f_k, l_k)]$	a port with k fibres running along links
Fibre	$\ni$	f	::=	Demand d	demand d uses this entire fibre
			|	Bands $[b_1, \ldots, b_n]$	n band details for this fibre
Band	$\ni$	b	::=	Demand d	demand d uses this entire band
			|	Waves $[w_1, \ldots, w_{m/n}]$	m/n wavelength details for this band
Wave	$\ni$	w	::=	Demand d	demand d uses this wavelength
			|	None	this wavelength is unused

As a shorthand notation, for ports we use None to denote $[(\text{None}, \text{None}), \ldots, (\text{None}, \text{None})]$, for fibres we use None to denote Bands $[\text{None}, \ldots, \text{None}]$ and for bands we use None to denote Waves $[\text{None}, \ldots, \text{None}]$.

Furthermore, we extend the list indexing notation: for a port $p = [(f_1, l_1), \ldots, (f_k, l_k)]$ we have $p[i] = f_i$, for a fibre $f = \text{Bands } [b_1, \ldots, b_n]$, we have $f[i] = b_i$, and for a band $b = \text{Waves } [w_1, \ldots, w_{m/n}]$, we have $b[i] = w_i$.

We introduce the function 'usage' which returns all if all wavelengths of its argument are in use, and canonical to change a bandful or fibreful of wavelengths for the same demand into one band or fibre, respectively, for this demand:

$$
\begin{array}{lcl}
\mathrm{usage}^{\mathrm{f}}(\mathsf{Demand}\ d) & = & \mathsf{all} \\
\mathrm{usage}^{\mathrm{f}}(\mathsf{Bands}\ [b_1,\ldots,b_n]) & = & \mathsf{all},\ \textbf{if}\ \forall i : \mathrm{usage}^{\mathrm{b}}(b_i) = \mathsf{all} \\
\mathrm{usage}^{\mathrm{f}}(\mathsf{Bands}\ [b_1,\ldots,b_n]) & = & \mathsf{Bands}\ [\mathrm{usage}^{\mathrm{b}}(b_1),\ldots,\mathrm{usage}^{\mathrm{b}}(b_n)],\ \textbf{otherwise} \\
\mathrm{usage}^{\mathrm{b}}(\mathsf{Demand}\ d) & = & \mathsf{all} \\
\mathrm{usage}^{\mathrm{b}}(\mathsf{Waves}\ [\mathsf{Demand}\ d_1,\ldots,\mathsf{Demand}\ d_{m/n}]) & & \\
 & = & \mathsf{all} \\
\mathrm{usage}^{\mathrm{b}}(\mathsf{Waves}\ [w_1,\ldots,w_{m/n}]) & = & \mathsf{Waves}\ [w_1,\ldots,w_{m/n}],\ \textbf{otherwise}
\end{array}
$$

$$
\begin{array}{lcl}
\mathrm{canonical}^{\mathrm{f}}(\mathsf{Demand}\ d) & = & \mathsf{Demand}\ d \\
\mathrm{canonical}^{\mathrm{f}}(\mathsf{Bands}\ [b_1,\ldots,b_n]) & = & \mathsf{Demand}\ d,\ \textbf{if}\ \forall i : \mathrm{canonical}^{\mathrm{b}}(b_i) = \mathsf{Demand}\ d \\
\mathrm{canonical}^{\mathrm{f}}(\mathsf{Bands}\ [b_1,\ldots,b_n]) & = & \mathsf{Bands}\ [b'_1,\ldots,b'_n]),\ \textbf{otherwise} \\
\qquad\qquad \textbf{where}\ \forall i : b'_i & = & \mathrm{canonical}^{\mathrm{b}}(b_i) \\
\mathrm{canonical}^{\mathrm{b}}(\mathsf{Waves}\ [\mathsf{Demand}\ d,\ldots,\mathsf{Demand}\ d]) & & \\
 & = & \mathsf{Demand}\ d, \\
\mathrm{canonical}^{\mathrm{b}}(b) & = & b,\ \textbf{otherwise}
\end{array}
$$

So for instance we find that

$$\mathrm{canonical}^{\mathrm{f}}(\mathsf{Bands}\ [\mathsf{Waves}\ [\mathsf{Demand}\ 42, \mathsf{Demand}\ 42], \mathsf{Demand}\ 42]) = \mathsf{Demand}\ 42.$$

If two ports are mutually exclusive, that is, if no wavelength / fibre is in use for both ports, they can be or'ed together:

$$
\begin{array}{lcll}
\mathrm{or}^{\mathrm{p}}([f_1,\ldots,f_k],[f'_1,\ldots,f'_k]) & = & [\mathrm{or}^{\mathrm{f}}(f_1,f'_1),\ldots,\mathrm{or}^{\mathrm{f}}(f_k,f'_k)] & \\
\mathrm{or}^{\mathrm{f}}(f,f') & = & f, & \textbf{if}\ \mathrm{usage}^{\mathrm{f}}(f') = \mathsf{none} \\
\mathrm{or}^{\mathrm{f}}(f,f') & = & f', & \textbf{if}\ \mathrm{usage}^{\mathrm{f}}(f) = \mathsf{none} \\
\mathrm{or}^{\mathrm{f}}(\mathsf{Bands}\ [b_1,\ldots,b_n],\mathsf{Bands}\ [b'_1,\ldots,b'_n]) & & & \\
\quad = \mathsf{Bands}\ [\mathrm{or}^{\mathrm{b}}(b_1,b'_1),\ldots,\mathrm{or}^{\mathrm{b}}(b_n,b'_n)], & & & \textbf{otherwise} \\
\mathrm{or}^{\mathrm{b}}(b,b') & = & b, & \textbf{if}\ \mathrm{usage}^{\mathrm{b}}(b') = \mathsf{none} \\
\mathrm{or}^{\mathrm{b}}(b,b') & = & b', & \textbf{if}\ \mathrm{usage}^{\mathrm{b}}(b) = \mathsf{none} \\
\mathrm{or}^{\mathrm{b}}(\mathsf{Waves}\ [w_1,\ldots,w_{m/n}],\mathsf{Waves}\ [w'_1,\ldots,w'_{m/n}]) & & & \\
\quad = \mathsf{Waves}\ [\mathrm{or}^{\mathrm{w}}(w_1,w'_1),\ldots,\mathrm{or}^{\mathrm{w}}(w_{m/n},w'_{m/n})], & & & \textbf{otherwise} \\
\mathrm{or}^{\mathrm{w}}(\mathsf{None},w) & = & w & \\
\mathrm{or}^{\mathrm{w}}(w,\mathsf{None}) & = & w & \\
\mathrm{or}^{\mathrm{w}}(\mathsf{Demand}\ d_1,\mathsf{Demand}\ d_2) & = & \mathsf{error} &
\end{array}
$$

So, for instance we have the following port calculation:

p_1
$= [(\text{Demand } 42, l_7),$
$\quad ([\text{None},$
$\quad\ \text{None},$
$\quad\ \text{None},$
$\quad\ [\text{Demand } 42, \text{Demand } 42],$
$\quad\ \text{Demand } 42,$
$\quad\ [\text{None}, \text{Demand } 42],$
$\quad\ \text{None},$
$\quad\ \text{None}], l_5)]$

p_2
$= [(\text{None}, l_7),$
$\quad ([\text{Demand } 23,$
$\quad\ \text{None},$
$\quad\ [\text{None}, \text{Demand } 17],$
$\quad\ \text{None},$
$\quad\ \text{None},$
$\quad\ [\text{Demand } 23, \text{None}],$
$\quad\ \text{Demand } 17,$
$\quad\ \text{None}], l_5)]$

$\text{or}^{\text{p}}(p_1, p_2)$
$= [(\text{Demand } 42, l_7),$
$\quad ([\text{Demand } 23,$
$\quad\ \text{None},$
$\quad\ [\text{None}, \text{Demand } 17],$
$\quad\ [\text{Demand } 42, \text{Demand } 42],$
$\quad\ \text{Demand } 42,$
$\quad\ [\text{Demand } 23, \text{Demand } 42],$
$\quad\ \text{Demand } 17,$
$\quad\ \text{None}], l_5)]$

for two ports p_1 and p_2, each with $k = 2$ fibres running along links l_7 and l_5, each fibre with $n = 8$ bands and each band with $m/n = 2$ wavelengths.

Several helper functions are used in the algorithms:

$\text{inlink}(\pi, s)$	Returns the link in π whose destination switch is s.
$\text{outtoin}(p_{out}, l)$	Returns the p_{in} corresponding to p_{out}, assuming link l connects p_{out} to p_{in}.
$\text{findorallocatefree}^{\text{f}}(p^{used})$	Finds a free fibre, allocating a new fibre if necessary.
$\text{findorallocatefree}^{\text{b}}(p^{used}, l)$	Finds a free band on link l, allocating a new fibre if necessary.
$\text{findorallocatefree}^{\text{w}}(p^{used}, l)$	Finds a free wavelength on link l, allocating a new fibre if necessary.
$\text{findfreesingle}^{\text{w}}(p^{used}, l)$	Finds a free wavelength on link l in p^{used} in a band occupied by at least one other wavelength, or return nil if not possible.

$\text{switch}^{\text{f}}(s, f_{in}, f_{out})$	Add switch information to s, switching fibre f_{in} to f_{out}.
$\text{switch}^{\text{b}}(s, \langle f_{in}, b_{in}\rangle, \langle f_{out}, b_{out}\rangle)$	Add switch information to s, switching band b_{in} on fibre f_{in} to band b_{out} on fibre f_{out}.
$\text{switch}^{\text{w}}(s, \langle f_{in}, b_{in}, w_{in}\rangle, \langle f_{out}, b_{out}, w_{out}\rangle)$	Add switch information to s, switching wavelength w_{in} of band b_{in} on fibre f_{in} to wavelength w_{out} of band b_{out} on fibre f_{out}.

A.2 Calculating the first output port

The algorithm for calculating the first input port at node s_0 is shown in Figure A.1. We

```
findfirstport([π_1, ..., π_k], s_0) =
  p ← None
  p_temp ← inport(s_0)
  for i ← 1, ..., k do
    if inport(π_i, s_0) ≠ nil then
      p ← or^p(p, inport(π_i, s_0))
    else
      l ← inlink(π_i, s_0)
      p ← or^p(p, findfreeports(p_temp, v(π_i), l))
  return p
```

Figure A.1: Finding the first input port

take the ports from all the paths that have already been routed up to s_0, and for the remaining paths we just add a minimally-packed port of the correct volume v, constructed by function findfreeports(p^{used}, v) defined in Figure A.2.

Assume given a path $\pi = [l_1, \ldots, l_k]$ along which a volume v of wavelengths must be allocated, and assume that they have already been allocated along the first i links. Let p_i^{π} be a port describing the wavelengths / bands / fibres that π uses on link i connected to the input port of s_i, and let p_{i+1}^{used} be a port describing the wavelengths / bands on the output port of s_i that cannot be used because they are already occupied. We now determine p_{i+1}^{π}, and update s_i and p_{i+1}^{used} to reflect the assigned port usage, by the algorithm shown in Figures A.3–A.7. Variable j^{b} is used to index into arrays f, b and w: each time a new band is discovered in the input port, its input port address (i.e. its fibre and band number) is stored in $f[j^{\text{b}}]$ and $b[j^{\text{b}}]$, and j^{b} is incremented. When n bands have been discovered, a new fibre is created in the output port. Then the switching of the n bands is performed,

findfreeports(p^{used}, v, l) =
  while $v \geq m$ **do**
    $i^{\mathrm{f}} \leftarrow \mathrm{findorallocatefree}^{\mathrm{f}}(p^{used})$
    $p[i^{\mathrm{f}}] \leftarrow (\mathsf{all}, l)$
    $v \leftarrow v - m$
  while $v \geq m/n$ **do**
    $(i^{\mathrm{f}}, i^{\mathrm{b}}) \leftarrow \mathrm{findorallocatefree}^{\mathrm{b}}(p^{used}, l)$
    $(p[i^{\mathrm{f}}])[i^{\mathrm{b}}] \leftarrow \mathsf{all}$
    $v \leftarrow v - m/n$
  while $v \geq 1$ **do**
    $(i^{\mathrm{f}}, i^{\mathrm{b}}, i^{\mathrm{w}}) \leftarrow \mathrm{findorallocatefree}^{\mathrm{w}}(p^{used}, l)$
    $((p[i^{\mathrm{f}}])[i^{\mathrm{b}}])[i^{\mathrm{w}}] \leftarrow 1$
    $v \leftarrow v - 1$
  return p

Figure A.2: Finding free ports when adding wavelengths

assignoutputports($p_i^{\pi}, s_i, p_{i+1}^{used}, l_{i+1}$) =
  $p_{in} \leftarrow p_i^{\pi}$, $p_{out} \leftarrow \mathsf{None}$
  Assign switching in s_i for all fully used fibres from p_{in} to p_{out},
    leaving indexes for remaining fibres with any band/wavelength usage in I_b
  $j^{\mathrm{b}} \leftarrow 1$
  Assign switching in s_i for all fully used bands in fibres indexed in I_b from p_{in} to p_{out},
    leaving indexes for remaining fibres with any wavelength usage in I_w
  $j^{\mathrm{w}} \leftarrow 1$
  Assign switching in s_i for remaining wavelengths in fibres indexed in I_w from p_{in} to p_{out}
  Fit as much of the last switchings in the fibre into free positions in p_{i+1}^{used},
    assigning a new fibre for any remaining wavelengths
  return (outtoin(canonical(p_{out}), l_{i+1}))

Figure A.3: Assigning output ports at a switch

```
for i_in^f ← 1,...,fibres(p_in) do
  f_in ← p_in[i_in^f]
  if usage^f(f_in) = all then
    i_out^f ← findorallocatefree^f(p_{i+1}^used)
    switch^f(s, i_in^f, i_out^f)
    p_out[i_out^f] ← f_in
  else
    if f_in ≠ None then I_b ← append(I_b, i_in^f)
```

Figure A.4: Step 3 in Figure A.3: Assigning switching of entire fibres

```
for i_in^f ∈ I_b do
  f_in ← p_in[i_in^f]
  for i_in^b ← 1,...,n do
    b_in ← f_in[i_in^b]
    if usage^b[b_in] = all then
      f[j^b] ← i_in^f,   b[j^b] ← i_in^b
      j^b ← j^b + 1
      if j^b > n then
        i_out^f ← findorallocatefree^f(p_{i+1}^used)
        bs ← [ ]
        for i_out^b ← 1,...,n do
          switch^b(s, ⟨f[i_out^b], b[i_out^b]⟩, ⟨i_out^f, i_out^b⟩)
          bs ← append(bs, p_in[f[i_out^b]][b[i_out^b]])
        p_out[i_out^f] ← Bands bs
        j^b ← 1
    else
      if b_in ≠ None then I_w ← append(I_w, (i_in^f, i_in^b))
```

Figure A.5: Step 5 in Figure A.3: Assigning switching of entire bands

based on their input port addresses stored in the f and b arrays, and j^{b} is reset to 1. The process when individual wavelengths are discovered (cf. Figure A.6) is analogous,

```
for (i_in^f, i_in^b) ∈ I_w do
  f_in ← p_in[i_in^f],   b_in ← f_in[i_in^b]
  for i_in^w ← 1,...,m/n do
    if b_in[i_in^w] = Demand d then
      b[j^b] ← nil,   f[j^b, j^w] ← i_in^f,   b[j^b, j^w] ← i_in^b,   w[j^b, j^w] ← i_in^w
      j^w ← j^w + 1
      if j^w > m/n then
        j^b ← j^b + 1
        if j^b > n then
          i_out^f ← findorallocatefree^f[p_{i+1}^used]
          bs ← [ ]
          for i_out^b ← 1,...,n do
            if b[i_out^b] ≠ nil then
              switch^b(s, ⟨f[i_out^b], b[i_out^b]⟩, ⟨i_out^f, i_out^b⟩)
              bs ← append(bs, p_in[f[i_out^b]][b[i_out^b]])
            else
              ws ← [ ]
              for i_out^w ← 1,...,m/n do
                switch^w(s, ⟨f[i_out^b, i_out^w], b[i_out^b, i_out^w], w[i_out^b, i_out^w]⟩, ⟨i_out^f, i_out^b, i_out^w⟩)
                ws ← append(ws, p_in[f[i_out^b, i_out^w]][b[i_out^b, i_out^w]][w[i_out^b, i_out^w]])
              bs ← append(bs, Waves ws)
          p_out[i_out^f] ← Bands bs
          j^b ← 1
        j^w ← 1
```

Figure A.6: Step 7 in Figure A.3: Assigning switching of individual wavelengths

making use also of j^{w} to index into all three arrays. This way, at most one fibre-full of input connections are pending at any time. Finally, the remaining pending bands and wavelengths are packed into any unused wavelengths in p_{i+1}^{used}, as shown in Figure A.7.

$p_{i+1}^{used\prime} \leftarrow p_{i+1}^{used}$
if $j^{\mathrm{b}} > 1 \vee j^{\mathrm{w}} > 1$ /* some bands and/or wavelengths remain */

/* first we fit single wavelengths into semi-full free output bands */
for $i_{out}^{\mathrm{b}} \leftarrow 1, \ldots, j^{\mathrm{b}}$ **do**
if $b[i_{out}^{\mathrm{b}}] = \mathsf{nil}$ **then**
if $i_{out}^{\mathrm{b}} = j^{\mathrm{b}}$ **then** $i^{w\prime} \leftarrow j^{\mathrm{w}}$ **else** $i^{w\prime} = m/n$
for $i_{out}^{\mathrm{w}} \leftarrow 1, \ldots, i^{w\prime}$ **do**
$(i_{free}^{\mathrm{f}}, i_{free}^{\mathrm{b}}, i_{free}^{\mathrm{w}}) \leftarrow \mathrm{findfreesingle}^{\mathrm{w}}(p_{i+1}^{used\prime})$
if $i_{free}^{\mathrm{f}} \neq \mathsf{nil}$ **then** /* single free wavelength found */
$\mathrm{switch}^{\mathrm{w}}(s, \langle f[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}], b[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}], w[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}]\rangle, \langle i_{free}^{\mathrm{f}}, i_{free}^{\mathrm{b}}, i_{free}^{\mathrm{w}}\rangle)$
$p_{out}[i_{free}^{\mathrm{f}}][i_{free}^{\mathrm{b}}][i_{free}^{\mathrm{w}}] \leftarrow p_{in}[f[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}]][b[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}]][w[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}]]$
else
$I_{rest} \leftarrow \mathrm{append}(I_{rest}, (i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}))$
else
$I_{rest} \leftarrow \mathrm{append}(I_{rest}, (i_{out}^{\mathrm{b}}, \mathsf{nil}))$

/* then we fit remaining bands into free output bands, */
/* and allocate a new fibre if necessary */
for $(i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}) \in I_{rest}$ **do**
if $b[i_{out}^{\mathrm{b}}] \neq \mathsf{nil}$ **then**
$(i_{free}^{\mathrm{f}}, i_{free}^{\mathrm{b}}) \leftarrow \mathrm{findorallocatefree}^{\mathrm{b}}(p_{i+1}^{used\prime})$
$\mathrm{switch}^{\mathrm{b}}(s, \langle f[i_{out}^{\mathrm{b}}], b[i_{out}^{\mathrm{b}}], w[i_{out}^{\mathrm{b}}]\rangle, \langle i_{free}^{\mathrm{f}}, i_{free}^{\mathrm{b}}\rangle)$
$p_{out}[i_{free}^{\mathrm{f}}][i_{free}^{\mathrm{b}}] \leftarrow p_{in}[f[i_{out}^{\mathrm{b}}]][b[i_{out}^{\mathrm{b}}]]$
else
$I'_{rest} \leftarrow \mathrm{append}(I'_{rest}, (i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}))$

/* finally, we allocate remaining single wavelengths */
for $(i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}) \in I'_{rest}$ **do**
$(i_{free}^{\mathrm{f}}, i_{free}^{\mathrm{b}}, i_{free}^{\mathrm{w}}) \leftarrow \mathrm{findorallocatefree}^{\mathrm{w}}(p_{i+1}^{used\prime})$
$\mathrm{switch}^{\mathrm{w}}(s, \langle f[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}], b[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}], w[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}]\rangle, \langle i_{free}^{\mathrm{f}}, i_{free}^{\mathrm{b}}, i_{free}^{\mathrm{w}}\rangle)$
$p_{out}[i_{free}^{\mathrm{f}}][i_{free}^{\mathrm{b}}][i_{free}^{\mathrm{w}}] \leftarrow p_{in}[f[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}]][b[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}]][w[i_{out}^{\mathrm{b}}, i_{out}^{\mathrm{w}}]]$

Figure A.7: Step 8 in Figure A.3: Fitting remaining switchings into free wavelengths

A.3 Joining partially routed paths

When it is discovered that a subpath just routed has joined a subpath of π to an already routed subpath in node s (like in Step 4 of Figure 3.10), the algorithm shown in Figures A.8 and A.9 is used to actually set up switch s for the demand d that is satisfied by path π, switching connections for demand d from p_{in} to p_{out}.

A.4 Updating Σ

This updating of Σ is accomplished by the algorithm shown in Figure A.10.

A.5 Setting up switches for adding and dropping

Adding and dropping fibres, bands and wavelengths at a node is very similar to finding the next output port; the details are shown in Figure A.11. An analogous algorithm where "out" and "in" are reversed is used to set up the switches for adding.

```
makeswitching(s, p_in, p_out, Demand d) =
  fs ← [ ],   bs ← [ ],   ws ← [ ]
 for i^f_out ← 1, ..., fibres(p_out) do
  f ← p_out[i^f_out]
  if f = Demand d then
   fs ← append(fs, i^f_out)
  else if f ≠ None then
   for i^b_out ← 1, ..., n do
    b ← f[i^b_out]
    if b = Demand d then
     bs ← append(bs, ⟨i^f_out, i^b_out⟩)
    else if b ≠ None then
     for i^w_out ← 1, ..., m/n do
      if b[i^w_out] = Demand d then
       ws ← append(ws, ⟨i^f_out, i^b_out, i^w_out⟩)
 for i^f_in ← 1, ..., fibres(p_in) do
  f ← p_in[i^f_in]
  if f = Demand d then
   findfibre(s, i^f_in, fs, bs, ws)
  else
   I_b ← append(I_b, i^f_in)
 for i^f_in ∈ I_b do
  for i^b_in ← 1, ..., n do
   b ← p[i^f_in][i^b_in]
   if b = Demand d then
    findband(s, ⟨i^f_in, i^b_in⟩, fs, bs, ws)
   else
    I_w ← append(I_w, ⟨i^f_in, i^b_in⟩)
 for ⟨i^f, i^b⟩ ∈ I_w do
  for i^w_in ← 1, ..., m/n do
   w ← p[i^f_in][i^b_in][i^w_in]
   if w = Demand d then
    findwave(s, ⟨i^f_in, i^b_in, i^w_in⟩, fs, bs, ws)
```

Figure A.8: Switching demand d from input port to output port

```
findfibre(s, i^f_in, fs, bs, ws) =
  if not(empty(fs)) then
    switch^f(s, i^f_in, pop(fs))
  else
    for i^b_in ← 1, ..., n do findband(s, ⟨i^f_in, i^b_in⟩, fs, bs, ws)
findband(s, ⟨i^f_in, i^b_in⟩, fs, bs, ws) =
  if not(empty(bs)) then
    switch^b(s, ⟨i^f_in, i^b_in⟩, pop(bs))
  else if not(empty(fs)) then
    i^f_out ← pop(fs)
    for i^b_out ← 2, ..., n do append(bs, ⟨i^f_out, i^b_out⟩)
    switch^b(s, ⟨i^f_in, i^b_in⟩, ⟨i^f_out, 1⟩)
  else
    for i^w_in ← 1, ..., m/n do findwave(s, ⟨i^f_in, i^b_in, i^w_in⟩, fs, bs, ws)
findwave(s, ⟨i^f_in, i^b_in, i^w_in⟩, fs, bs, ws) =
  if not(empty(ws)) then
    switch^w(s, ⟨i^f_in, i^b_in, i^w_in⟩, pop(ws))
  else if not(empty(bs)) then
    ⟨i^f_out, i^b_out⟩ ← pop(bs)
    for i^w_out ← 2, ..., m/n do append(ws, ⟨i^f_out, i^b_out, i^w_out⟩)
    switch^w(s, ⟨i^f_in, i^b_in, i^w_in⟩, ⟨i^f_out, i^b_out, 1⟩)
  else if not(empty(fs)) then
    i^f_out ← pop(fs)
    for i^b_out ← 2, ..., n do append(bs, ⟨i^f_out, i^b_out⟩)
    for i^w_out ← 2, ..., m/n do append(ws, ⟨i^f_out, 1, i^w_out⟩)
    switch^w(s, ⟨i^f_in, i^b_in, i^w_in⟩, ⟨i^f_out, 1, 1⟩)
  else error
```

Figure A.9: Switching demand d from input port to output port

```
updatesubpathset(π, Π) =
  for l ∈ π do
    for (π′, Π′) ∈ subpaths(l) do
      if Π′ ∩ Π ≠ {} then            /* Only consider subpaths with common paths */
        Σ ← remove(Σ, (π′, Π′))
        if Π′ \ Π ≠ {} then Σ ← add(Σ, (π′, Π′ \ Π))        /* Add narrowed path */
        i′ ← length(π′)
        started ← false
        for i ← 1, ..., i′ do
          if π′(i) ∈ π then
            if started then Σ ← add(Σ, ([l_{i_start}, ..., l_{i−1}], Π′)) /* Add shortened path */
            started ← false
          else
            if not(started) then i_start ← i
            started ← true
        if started then Σ ← add(Σ, ([l_{i_start}, ..., l_{i′}], Π′))
```

Figure A.10: Updating the set Σ of subpaths after routing subpath (π, Π)

```
setupdrops() =
  for s ∈ Switch do
    p_temp ← None
    for π ← paths(inport(s)) do
      if isdestnode(s, π) then p_temp ← or^p(p_temp, inport(π, s))
    dropport(s) ← assignoutputports(p_temp, s, outport(s), nil)
```

Figure A.11: Setting up the switches for dropping

Appendix B

Empirical studies of analytic blocking calculations

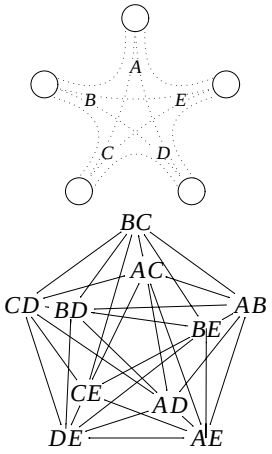
This appendix reports the results of some empirical studies of the differences in blocking between WDM and ES networks. We use the wdm2csn program described in Section 5 to convert ESN specifications (link / route tables) into WDM network specifications (constraint / route tables). Both kinds of tables are then used as input to the pincon program Pinsky and Conway (1992); Dickmeiss and Larsen (1993) to calculate the exact end-to-end blockings on the routes.

For each network, we show its routes, the constraint / route table that is computed by wdm2csn, and for simple cases the interference graph (i.e. the graph that shows which routes interfere with each other).

The networks are shown in Figures 5.10–5.11 and 5.12, and the traffic loads are described in Section 5.4.

B.1 Networks with 5 links

B.1.1 Star network with all-to-all routes

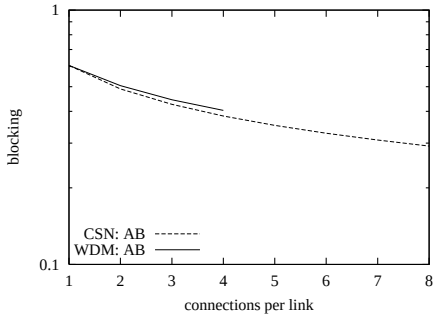


WDM network routes *AB*, *AC*, *AD*, *AE*, *BC*, *BD*, *BE*, *CD*, *CE* and *DE*

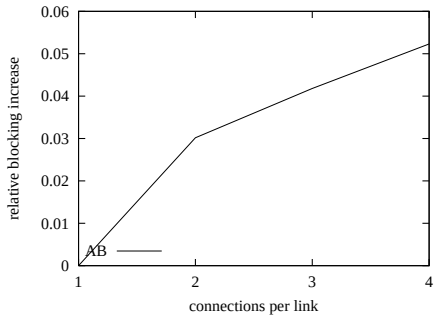
c	*AB*	*AC*	*AD*	*AE*	*BC*	*BD*	*BE*	*CD*	*CE*	*DE*
1	1	1	1	1	0	0	0	0	0	0
1	1	0	0	0	1	1	1	0	0	0
1	0	1	0	0	1	0	0	1	1	0
1	0	0	1	0	0	1	0	1	0	1
1	0	0	0	1	0	0	1	0	1	1
2	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	1	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0
1	1	0	0	1	0	0	1	0	0	0
1	0	1	1	0	0	0	0	1	0	0
1	0	1	0	1	0	0	0	0	1	0
1	0	0	1	1	0	0	0	0	0	1
1	0	0	0	0	1	1	0	1	0	0
1	0	0	0	0	1	0	1	0	1	0
1	0	0	0	0	0	1	1	0	0	1
1	0	0	0	0	0	0	0	1	1	1

Link balanced, path uniform & length dependent traffic

AB	*AC*	*AD*	*AE*	*BC*
0.250	0.250	0.250	0.250	0.250
BD	*BE*	*CD*	*CE*	*DE*
0.250	0.250	0.250	0.250	0.250

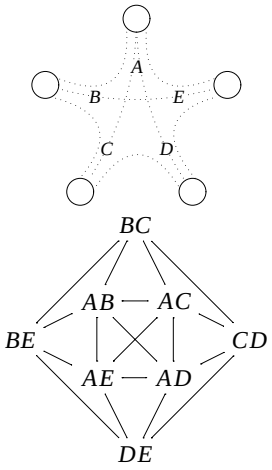


Absolute blocking in ESN and WDM networks—link balanced traffic (5starA-lb-blck)



Relative blocking differences between ESN and WDM networks—link balanced traffic (5starA-lb-diff)

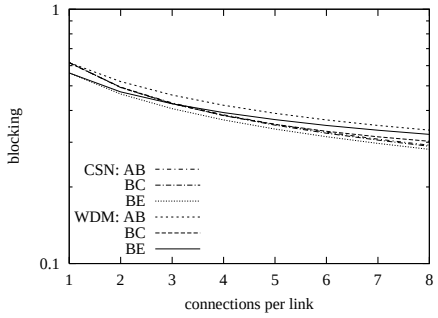
B.1.2 Star network with asymmetric route pattern



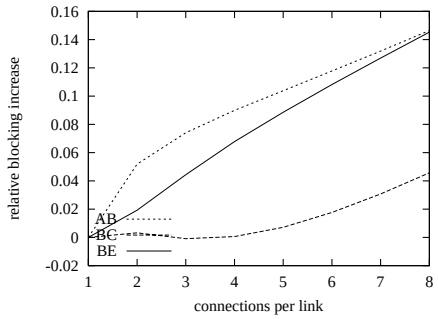
c	AB	AC	AD	AE	BC	BE	CD	DE
1	1	1	1	1	0	0	0	0
1	1	0	0	0	1	1	0	0
1	0	1	0	0	1	0	1	0
1	0	0	1	0	0	0	1	1
1	0	0	0	1	0	1	0	1
2	1	1	1	1	1	1	1	1
1	1	1	0	0	1	0	0	0
1	1	0	0	1	0	1	0	0
1	0	1	1	0	0	0	1	0
1	0	0	1	1	0	0	0	1

Link balanced traffic

AB	AC	AD	AE
0.250	0.250	0.250	0.250
BC	**BE**	**CD**	**DE**
0.250	0.500	0.500	0.250



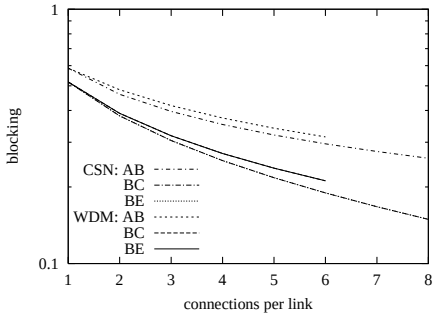
Absolute blocking in ESN and WDM networks—link balanced traffic (5starB-lb-blck)



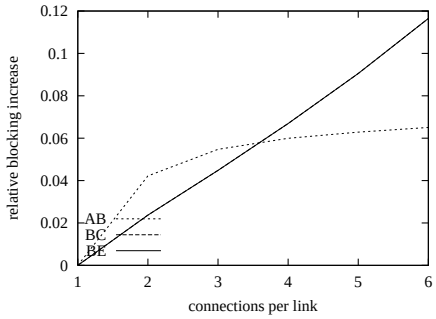
Relative blocking differences between ESN and WDM networks—link balanced traffic (5starB-lb-diff)

Path uniform & length dependent traffic

AB	AC	AD	AE
0.250	0.250	0.250	0.250
BC	**BE**	**CD**	**DE**
0.250	0.250	0.250	0.250

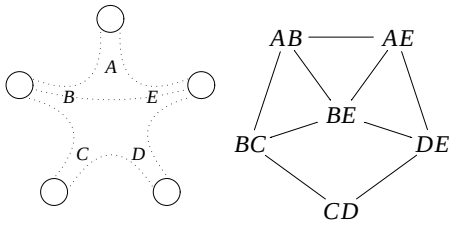


Absolute blocking in ESN and WDM networks—path uniform traffic
(5starB-pu-blck)



Relative blocking differences between ESN and WDM networks—path uniform traffic (5starB-pu-diff)

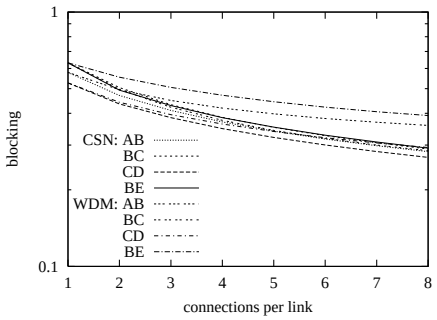
B.1.3 Star network with slightly asymmetric route pattern



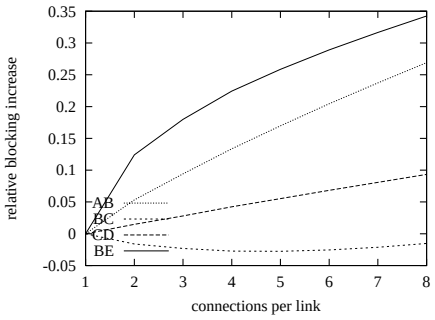
c	*AB*	*BC*	*CD*	*DE*	*AE*	*BE*
1	1	1	0	0	0	1
1	0	1	1	0	0	0
1	0	0	1	1	0	0
1	0	0	0	1	1	1
2	1	1	1	1	1	1
1	1	0	0	0	1	1

Link balanced traffic

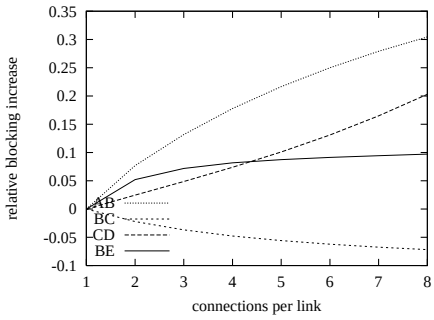
AB	*BC*	*CD*	*DE*	*AE*	*BE*
0.500	0.250	0.750	0.250	0.500	0.250



Absolute blocking in ESN and WDM networks—link balanced traffic
(5starC-lb-blck)



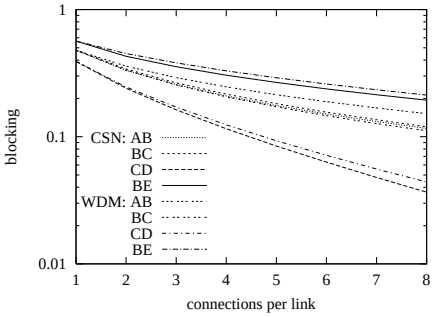
Relative blocking differences between ESN and WDM networks—link balanced traffic (5starC-lb-diff)



Relative blocking differences between ESN and WDM networks—path uniform traffic (5starC-pu-diff)

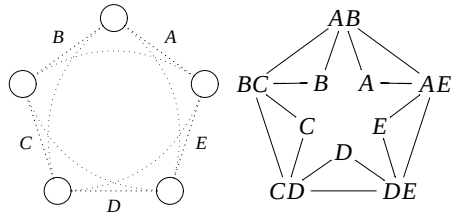
Path uniform & length dependent traffic

AB	*BC*	*CD*	*DE*	*AE*	*BE*
0.250	0.250	0.250	0.250	0.250	0.250



Absolute blocking in ESN and WDM networks—path uniform traffic (5starC-pu-blck)

B.1.4 Ring network with all-to-all routes



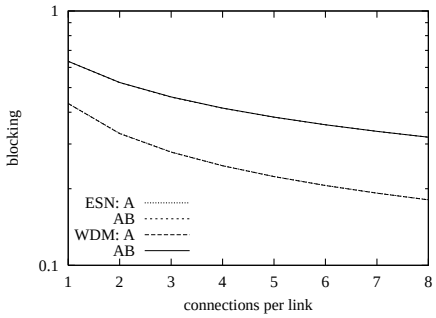
WDM network routes *A*, *B*, *C*, *D*, *E*, *AB*, *BC*, *CD*, *DE* and *AE*

c	*A*	*B*	*C*	*D*	*E*	*AB*	*BC*	*CD*	*DE*	*AE*
1	1	0	0	0	0	1	0	0	0	1
1	0	1	0	0	0	1	1	0	0	0
1	0	0	1	0	0	0	1	1	0	0
1	0	0	0	1	0	0	0	1	1	0
1	0	0	0	0	1	0	0	0	1	1
2	0	0	0	0	0	1	1	1	1	1

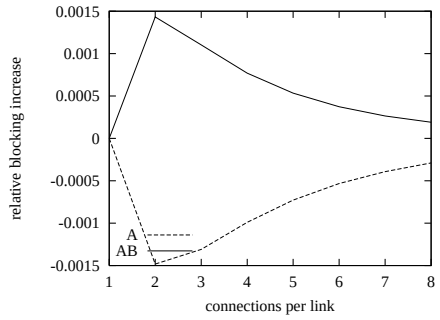
Link balanced & path uniform traffic

A	*B*	*C*	*D*	*E*
0.333	0.333	0.333	0.333	0.333

AB	*BC*	*CD*	*DE*	*AE*
0.333	0.333	0.333	0.333	0.333



Absolute blocking in ESN and WDM networks—link balanced traffic
(5ringA-lb-blck)

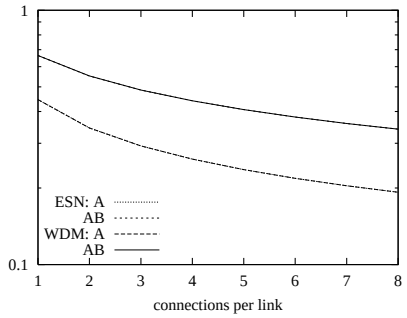


Relative blocking differences between ESN and WDM networks—link balanced traffic (5ringA-lb-diff)

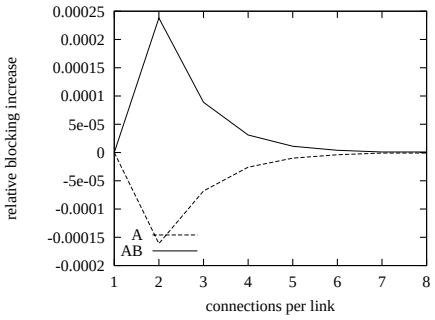
Length dependent traffic

A	*B*	*C*	*D*	*E*
0.500	0.500	0.500	0.500	0.500

AB	*BC*	*CD*	*DE*	*AE*
0.250	0.250	0.250	0.250	0.250

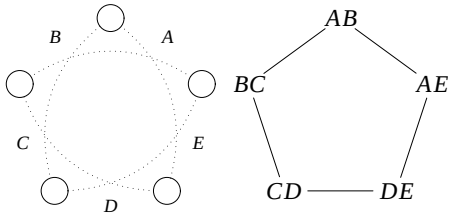


Absolute blocking in ESN and WDM networks—length-dependent traffic
(5ringA-ld-blck)



Relative blocking differences between ESN and WDM networks—length-dependent traffic
(5ringA-ld-diff)

B.1.5 Ring network with routes of length 2

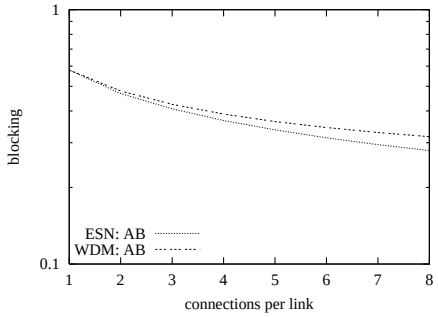


WDM network routes *AB*, *BC*, *CD*, *DE* and *AE*

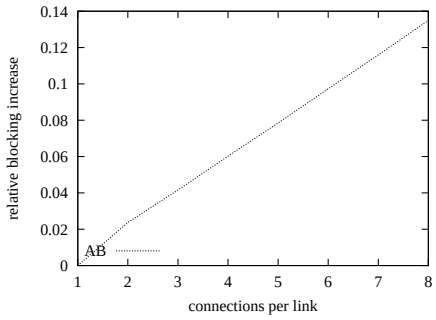
c	*AB*	*BC*	*CD*	*DE*	*AE*
1	1	0	0	0	1
1	1	1	0	0	0
1	0	1	1	0	0
1	0	0	1	1	0
1	0	0	0	1	1
2	1	1	1	1	1

Link balanced, path uniform & length dependent traffic

AB	*BC*	*CD*	*DE*	*AE*
0.500	0.500	0.500	0.500	0.500

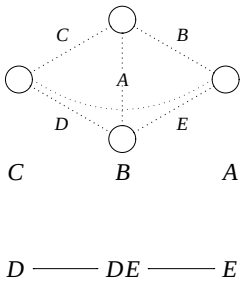


Absolute blocking in ESN and WDM networks—link balanced traffic
(5ringB-lb-blck)



Relative blocking differences between ESN and WDM networks—link balanced traffic (5ringB-lb-diff)

B.1.6 Double-ring network with all-to-all routes

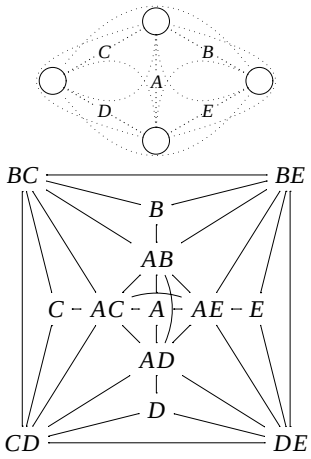


WDM network routes *A*, *B*, *C*, *D*, *E* and *DE*

c	*A*	*B*	*C*	*D*	*E*	*DE*
1	1	0	0	0	0	0
1	0	1	0	0	0	0
1	0	0	1	0	0	0
1	0	0	0	1	0	1
1	0	0	0	0	1	1

It is obvious from the interference graph that there is no difference in the constraint / route tables, and thus the blocking of the WDM and ESN networks.

B.1.7 Double-ring network with routes of length 1 & 2



WDM network routes *A*, *B*, *C*, *D*, *E*, *AB*, *AC*, *AE*, *AD*, *BC*, *BE*, *CD* and *DE*

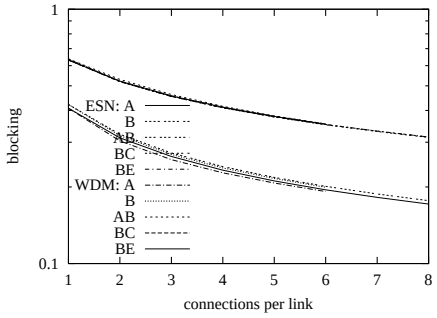
c	*A*	*B*	*C*	*D*	*E*	*AB*	*AC*	*AD*	*AE*	*BC*	*BE*	*CD*	*DE*
1	1	0	0	0	0	1	1	1	1	0	0	0	0
1	0	1	0	0	0	1	0	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	0	1	0	1	0
1	0	0	0	1	0	0	0	1	0	0	0	1	1
1	0	0	0	0	1	0	0	0	1	0	1	0	1
2	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	1	0	1	0	0
1	0	0	0	0	0	0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	0	1	1	0	0	0	1

Link balanced traffic

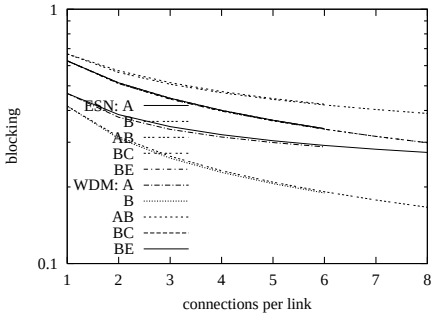
A	*B*	*C*	*D*	*E*
0.200	0.266	0.266	0.266	0.266

AB	*AC*	*AD*	*AE*
0.200	0.200	0.200	0.200

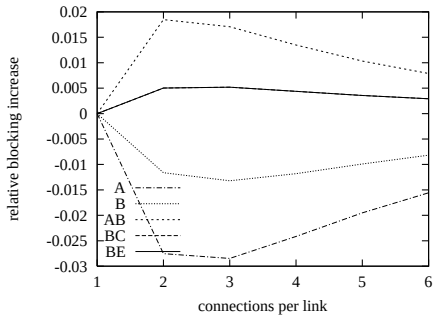
BC	*BE*	*CD*	*DE*
0.266	0.266	0.266	0.266



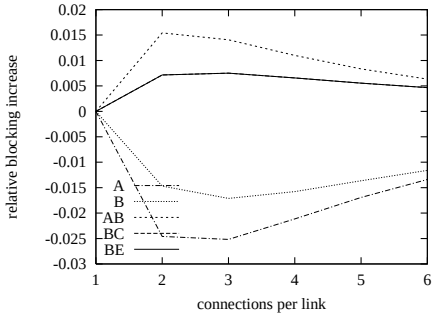
Absolute blocking in ESN and WDM networks—link balanced traffic (5dringB-lb-blck)



Absolute blocking in ESN and WDM networks—link balanced traffic (5dringB-pu-blck)



Relative blocking differences between ESN and WDM networks—link balanced traffic (5dringB-lb-diff)



Relative blocking differences between ESN and WDM networks—link balanced traffic (5dringB-pu-diff)

Path uniform traffic

A	*B*	*C*	*D*	*E*
0.250	0.250	0.250	0.250	0.250

AB	*AC*	*AD*	*AE*
0.250	0.250	0.250	0.250

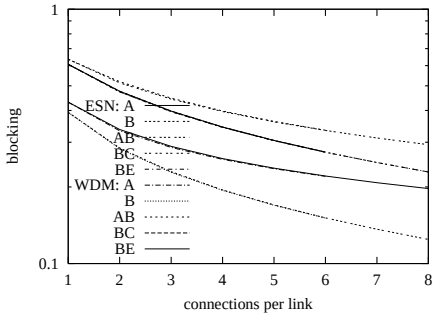
BC	*BE*	*CD*	*DE*
0.250	0.250	0.250	0.250

Length dependent traffic

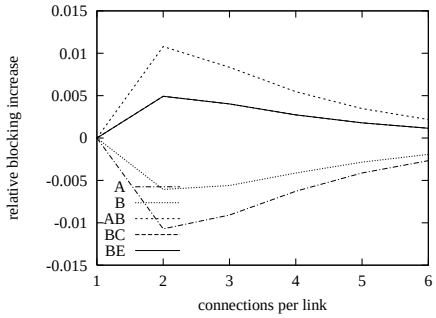
A	*B*	*C*	*D*	*E*
0.333	0.333	0.333	0.333	0.333

AB	*AC*	*AD*	*AE*
0.166	0.166	0.166	0.166

BC	*BE*	*CD*	*DE*
0.166	0.166	0.166	0.166



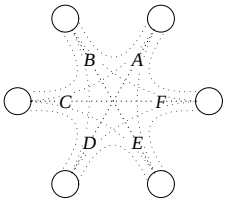
Absolute blocking in ESN and WDM networks—link balanced traffic
(5dringB-ld-blck)



Relative blocking differences between ESN and WDM networks—link balanced traffic (5dringB-ld-diff)

B.2 Networks with 6 links

B.2.1 Star network with all-to-all routes



WDM network routes *AB*, *AC*, *AD*, *AE*, *AF*, *BC*, *BD*, *BE*, *BF*, *CD*, *CE*, *CF*, *DE*, *DF* and *EF*

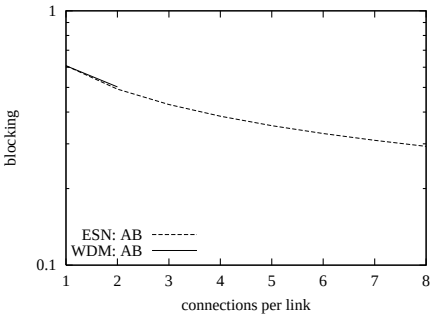
c	*AB*	*AC*	*AD*	*AE*	*AF*	*BC*	*BD*	*BE*	*BF*	*CD*	*CE*	*CF*	*DE*	*DF*	*EF*
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0
1	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1
1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1
2	1	1	1	0	1	1	1	0	1	1	0	1	0	1	0
1	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0
1	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2	1	1	1	1	0	1	1	1	0	1	1	0	1	0	0
1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0
2	1	0	1	1	1	0	1	1	1	0	0	0	1	1	1
2	1	1	0	1	1	1	0	1	1	0	1	1	0	0	1
1	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0
1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0
2	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1
1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0
1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0
1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0
2	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0

Link balanced, path uniform & length dependent traffic

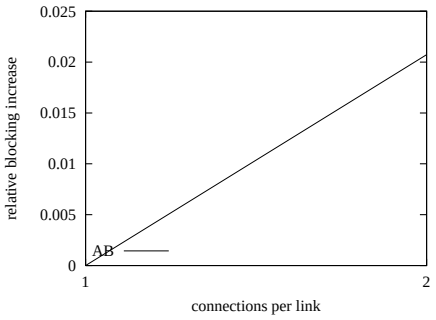
AB	*AC*	*AD*	*AE*	*AF*
0.200	0.200	0.200	0.200	0.200

BC	*BD*	*BE*	*BF*
0.200	0.200	0.200	0.200

CD	*CE*	*CF*	*DE*	*DF*	*EF*
0.200	0.200	0.200	0.200	0.200	0.200

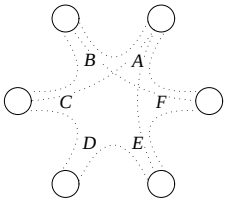


Absolute blocking in ESN and WDM networks—link balanced traffic
(6starA-lb-blck)



Relative blocking differences between ESN and WDM networks—link balanced traffic (6starA-lb-diff)

B.2.2 Star network with asymmetric route pattern

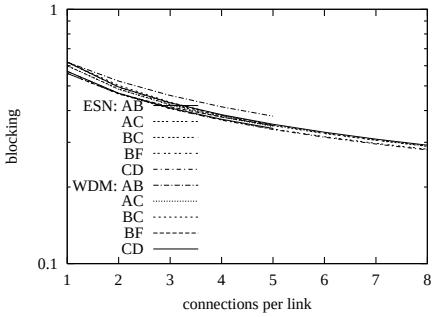


WDM network routes *AB*, *AC*, *AE*, *AF*, *BC*, *BF*, *CD*, *DE* and *EF*

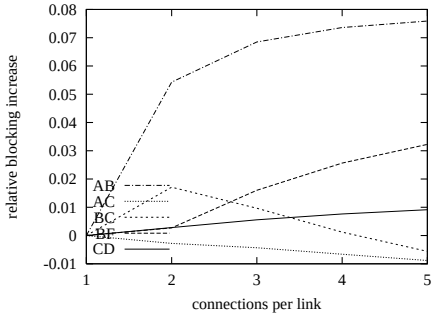
c	*AB*	*AC*	*AE*	*AF*	*BC*	*BF*	*CD*	*DE*	*EF*
1	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0
1	0	1	0	0	1	0	1	0	0
1	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0	1
2	1	1	1	1	1	1	0	0	1
2	1	1	1	0	1	0	1	1	0
1	1	0	0	1	0	1	0	0	0
1	1	1	0	0	1	0	0	0	0
2	0	1	1	1	0	0	1	1	1
1	0	0	1	1	0	0	0	0	1
2	0	0	0	0	1	1	1	1	1

Link balanced traffic

AB	*AC*	*AE*	*AF*	
0.250	0.250	0.250	0.250	
BC	*BF*	*CD*	*DE*	*EF*
0.250	0.500	0.500	0.500	0.250



Absolute blocking in ESN and WDM networks—link balanced traffic (6starB-lb-blck)

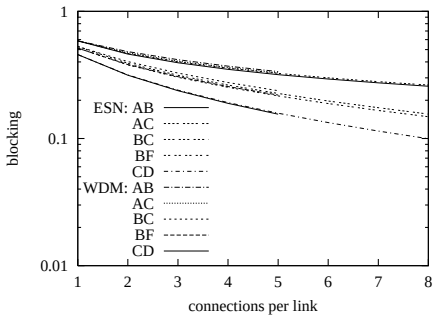


Relative blocking differences between ESN and WDM networks—link balanced traffic (6starB-lb-diff)

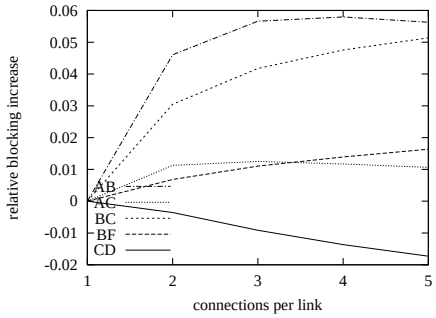
Path uniform & length dependent traffic

AB	*AC*	*AE*	*AF*
0.250	0.250	0.250	0.250

BC	*BF*	*CD*	*DE*	*EF*
0.250	0.250	0.250	0.250	0.250

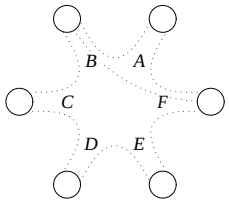


Absolute blocking in ESN and WDM networks—path uniform traffic (6starB-pu-blck)



Relative blocking differences between ESN and WDM networks—path uniform traffic (6starB-pu-diff)

B.2.3 Star network with slightly asymmetric route pattern

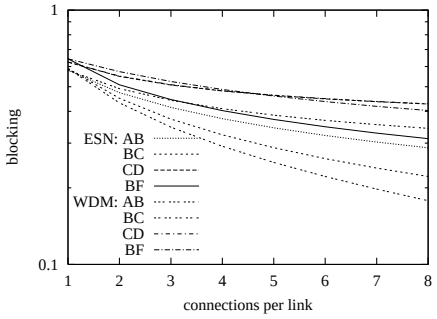


WDM network routes *AB*, *AF*, *BC*, *BF*, *CD*, *DE* and *EF*

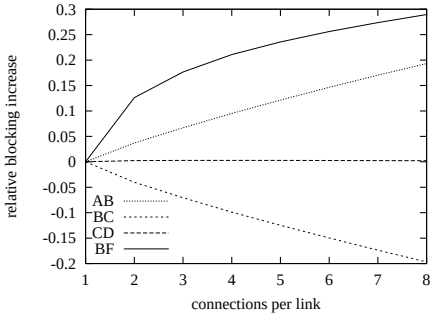
c	*AB*	*AF*	*BC*	*BF*	*CD*	*DE*	*EF*
1	1	0	1	1	0	0	0
1	0	0	1	0	1	0	0
1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	1
1	0	1	0	1	0	0	1
1	1	1	0	1	0	0	0
2	0	0	1	1	1	1	1

Link balanced traffic

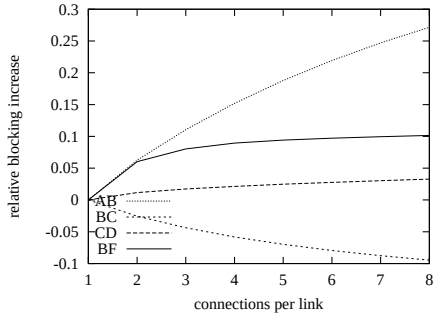
AB	*AF*	*BC*	*BF*	*CD*	*DE*	*EF*
0.250	0.250	0.250	0.500	0.500	0.500	0.250



Absolute blocking in ESN and WDM networks—link balanced traffic (6starC-lb-blck)



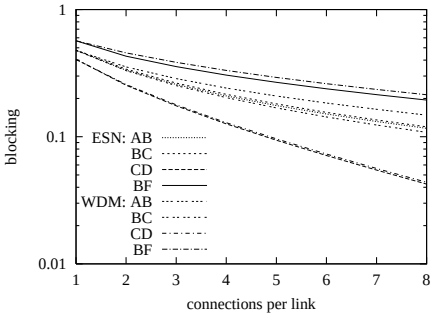
Relative blocking differences between ESN and WDM networks—link balanced traffic (6starC-lb-diff)



Relative blocking differences between ESN and WDM networks—path uniform traffic (6starC-pu-diff)

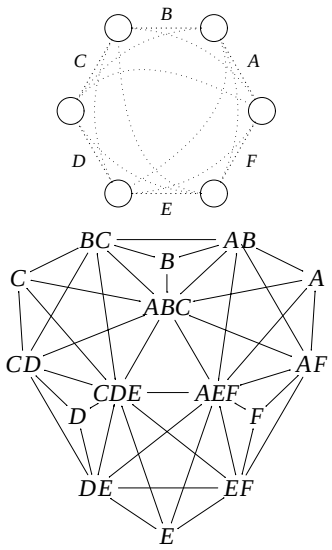
Path uniform & length dependent traffic

AB	*AF*	*BC*	*BF*	*CD*	*DE*	*EF*
0.250	0.250	0.250	0.250	0.250	0.250	0.250



Absolute blocking in ESN and WDM networks—path uniform traffic (6starC-pu-blck)

B.2.4 Ring network with all-to-all routes



WDM network routes *A*, *B*, *C*, *D*, *E*, *F*, *AB*, *BC*, *CD*, *DE*, *EF*, *AF*, *ABC*, *CDE*, and *AEF*

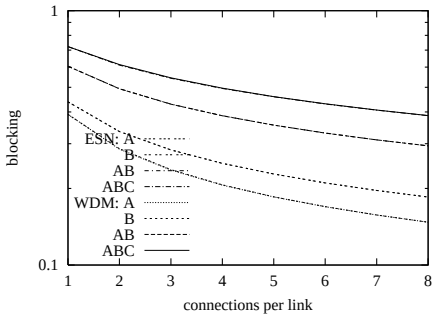
c	*A*	*B*	*C*	*D*	*E*	*F*	*AB*	*BC*	*CD*	*DE*	*EF*	*AF*	*ABC*	*CDE*	*AEF*
1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1
1	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0
1	0	0	1	0	0	0	0	1	1	0	0	0	1	1	0
1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	0
1	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1
2	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1
2	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1
2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Link balanced traffic

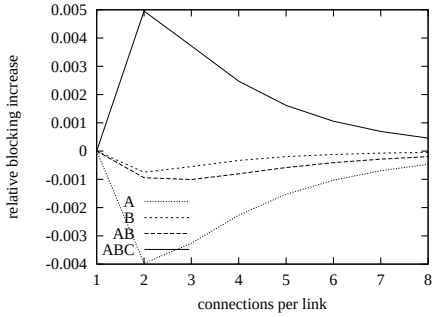
A	*B*	*C*	*D*	*E*	*F*
0.200	0.400	0.200	0.400	0.200	0.400

AB	*BC*	*CD*	*DE*	*EF*	*AF*
0.200	0.200	0.200	0.200	0.200	0.200

ABC	*CDE*	*AEF*
0.200	0.200	0.200



Absolute blocking in ESN and WDM networks—link balanced traffic (6ringA-lb-blck)



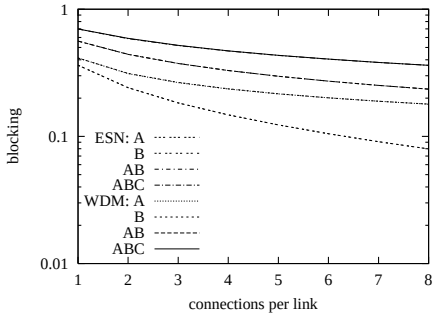
Relative blocking differences between ESN and WDM networks—link balanced traffic (6ringA-lb-diff)

Path uniform traffic

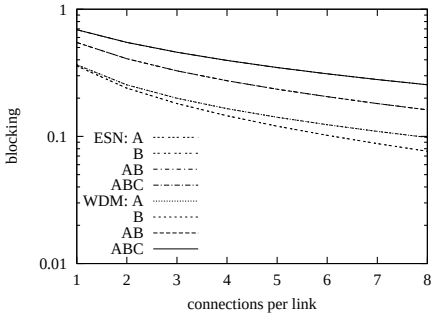
A	*B*	*C*	*D*	*E*	*F*
0.200	0.200	0.200	0.200	0.200	0.200

AB	*BC*	*CD*	*DE*	*EF*	*AF*
0.200	0.200	0.200	0.200	0.200	0.200

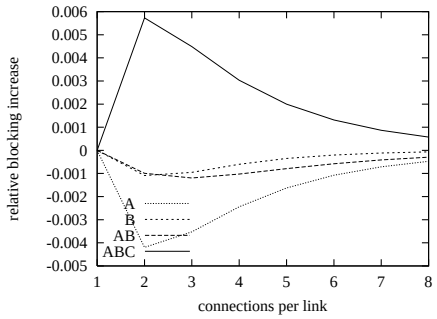
ABC	*CDE*	*AEF*
0.200	0.200	0.200



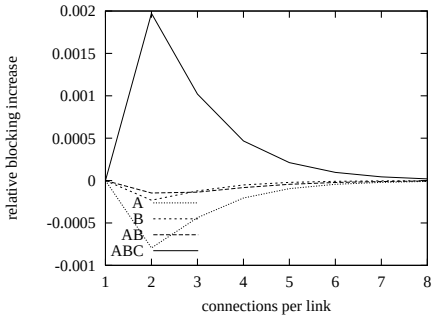
Absolute blocking in ESN and WDM networks—path uniform traffic
(6ringA-pu-blck)



Absolute blocking in ESN and WDM networks—length-dependent traffic
(6ringA-ld-blck)



Relative blocking differences between ESN and WDM networks—path uniform traffic (6ringA-pu-diff)

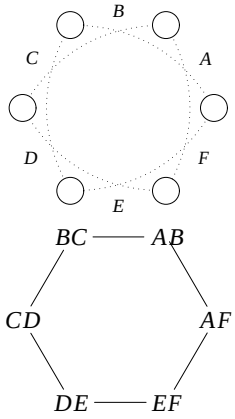


Relative blocking differences between ESN and WDM networks—length-dependent traffic
(6ringA-ld-diff)

Length dependent traffic

A	*B*	*C*	*D*	*E*	*F*
0.300	0.300	0.300	0.300	0.300	0.300
AB	*BC*	*CD*	*DE*	*EF*	*AF*
0.150	0.150	0.150	0.150	0.150	0.150
ABC	*CDE*	*AEF*			
0.100	0.100	0.100			

B.2.5 Ring network with routes of length 2

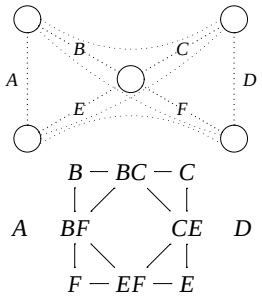


WDM network routes *AB*, *BC*, *CD*, *DE*, *EF* and *AF*

c	*AB*	*BC*	*CD*	*DE*	*EF*	*AF*
1	1	0	0	0	0	1
1	1	1	0	0	0	0
1	0	1	1	0	0	0
1	0	0	1	1	0	0
1	0	0	0	1	1	0
1	0	0	0	0	1	1

It is obvious from the constraint / route table that there is no difference in the blocking probabilities of the WDM and ESN networks.

B.2.6 Double-ring network with all-to-all routes

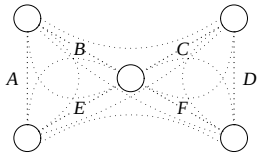


WDM network routes *A*, *B*, *C*, *D*, *E*, *F*, *BC*, *BF*, *CE* and *EF*

c	*A*	*B*	*C*	*D*	*E*	*F*	*BC*	*BF*	*CE*	*EF*
1	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	1	0
1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	1	1
1	0	0	0	0	0	1	0	1	0	1

It is obvious from the constraint / route table that there is no difference in the blocking probabilities of the WDM and ESN networks.

B.2.7 Double-ring network with routes of length 1 & 2



WDM network routes *A*, *B*, *C*, *D*, *E*, *F*, *AB*, *AE*, *BC*, *BE*, *BF*, *CD*, *CE*, *CF*, *DF* and *EF*

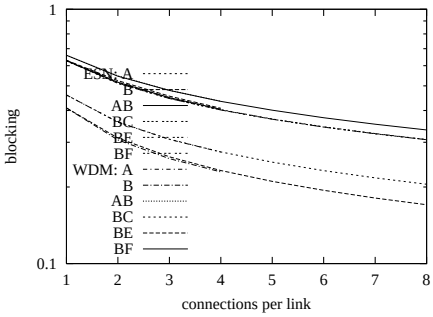
c	*A*	*B*	*C*	*D*	*E*	*F*	*AB*	*AE*	*BC*	*BE*	*BF*	*CD*	*CE*	*CF*	*DF*	*EF*
1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0	1	1	1	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1
1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1
2	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	
2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

Link balanced traffic

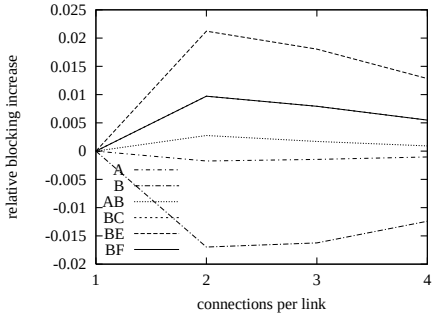
A	*B*	*C*	*D*	*E*	*F*
0.600	0.200	0.200	0.600	0.200	0.200

AB	*AE*	*BC*	*BE*	*BF*
0.200	0.200	0.200	0.200	0.200

CD	*CE*	*CF*	*DF*	*EF*
0.200	0.200	0.200	0.200	0.200



Absolute blocking in ESN and WDM networks—link balanced traffic (6dringB-lb-blck)



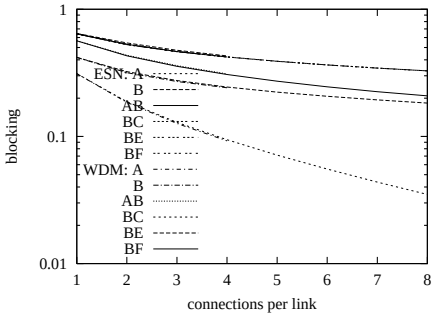
Relative blocking differences between ESN and WDM networks—link balanced traffic (6dringB-lb-diff)

Path uniform traffic

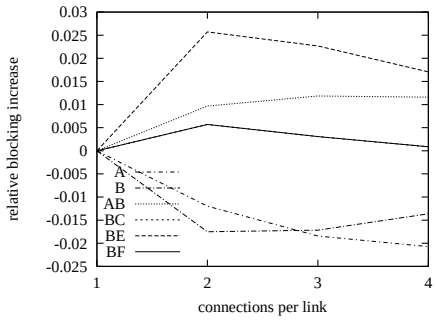
A	*B*	*C*	*D*	*E*	*F*
0.200	0.200	0.200	0.200	0.200	0.200

AB	*AE*	*BC*	*BE*	*BF*
0.200	0.200	0.200	0.200	0.200

CD	*CE*	*CF*	*DF*	*EF*
0.200	0.200	0.200	0.200	0.200



Absolute blocking in ESN and WDM networks—path uniform traffic
(6dringB-pu-blck)



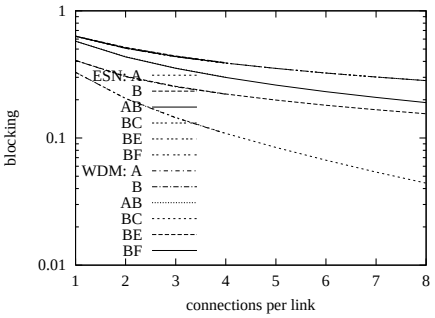
Relative blocking differences between ESN and WDM networks—path uniform traffic (6dringB-pu-diff)

Length dependent traffic

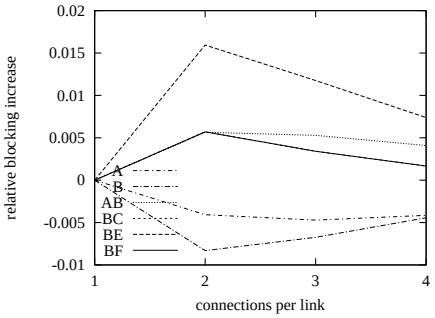
A	*B*	*C*	*D*	*E*	*F*
0.300	0.300	0.300	0.300	0.300	0.300

AB	*AE*	*BC*	*BE*	*BF*
0.150	0.150	0.150	0.150	0.150

CD	*CE*	*CF*	*DF*	*EF*
0.150	0.150	0.150	0.150	0.150



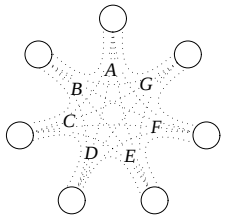
Absolute blocking in ESN and WDM networks—length dependent traffic
(6dringB-ld-blck)



Relative blocking differences between ESN and WDM networks—length dependent traffic (6dringB-ld-diff)

B.3 Networks with 7 links

B.3.1 Star Network with all-to-all routes

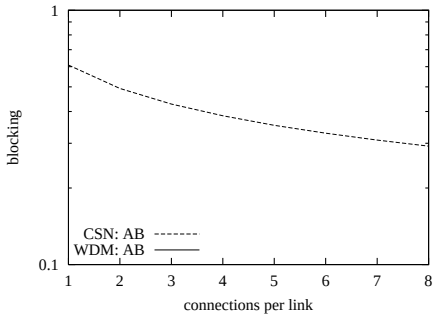


WDM network routes *AB*, *BC*, *CD*, *DE*, *EF*, *FG*, *AG*, *AD*, *BE*, *CF*, *DG*, *AE*, *BF*, *CG*, *AC*, *BD*, *CE*, *DF*, *EG*, *AF* and *BG*

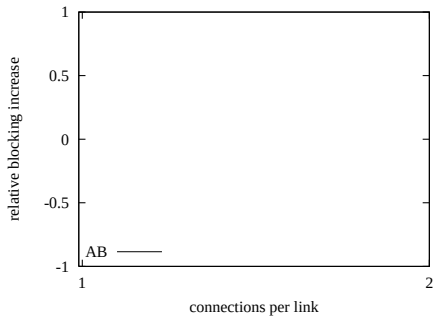
The results are shown in Figure B.1.

Link balanced, path uniform & length dependent traffic

AB	*BC*	*CD*	*DE*	*EF*	*FG*	*AG*
0.166	0.166	0.166	0.166	0.166	0.166	0.166
AD	*BE*	*CF*	*DG*	*AE*	*BF*	*CG*
0.166	0.166	0.166	0.166	0.166	0.166	0.166
AC	*BD*	*CE*	*DF*	*EG*	*AF*	*BG*
0.166	0.166	0.166	0.166	0.166	0.166	0.166



Absolute blocking in ESN and WDM networks—link balanced traffic (7starA-lb-blck)

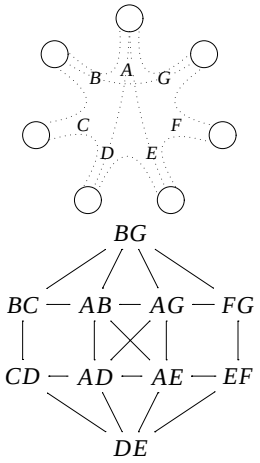


Relative blocking differences between ESN and WDM networks—link balanced traffic (7starA-lb-diff)

c	*AB*	*BC*	*CD*	*DE*	*EF*	*FG*	*AG*	*AD*	*BE*	*CF*	*DG*	*AE*	*BF*	*CG*	*AC*	*BD*	*CE*	*DF*	*EG*	*AF*	*BG*
1	1	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0
1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0
1	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	1	0
1	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	0
1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
2	1	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
2	1	1	1	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	0	1	0
1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
2	0	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	0	1	0
1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0	0	1	1
1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
2	0	0	1	1	1	0	0	1	0	1	0	1	0	0	1	0	1	1	0	1	0
1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0
2	1	1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
2	1	1	0	0	1	0	0	0	1	1	0	1	1	0	1	0	1	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1
2	1	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
2	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	0	0	0	1	1	1
2	0	1	1	1	0	0	0	0	1	0	1	0	1	0	0	1	0	1	0	1	1
2	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0
2	0	0	0	1	1	1	1	1	0	0	1	1	0	0	0	0	0	1	1	1	0
2	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	1	0	1	1	0	1
2	1	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	0	0	1	0	1
1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
2	0	1	1	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0	0
1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1
2	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	1	0	1	0	0	1
1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
2	0	0	1	1	1	1	0	0	0	1	1	0	0	1	0	0	1	1	1	0	0
2	0	0	1	1	0	0	1	1	0	0	1	1	0	1	1	0	1	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
2	0	0	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	0	1	0
1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Figure B.1: Constraint / route table for 7 link star WDM network

B.3.2 Star network with asymmetric route pattern



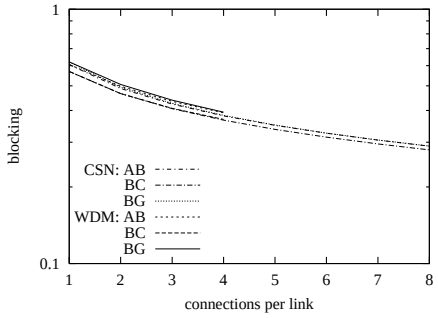
WDM network routes *AB*, *BC*, *CD*, *DE*, *EF*, *FG*, *AG*, *AD*, *AE*, and *BG*

c	*AB*	*BC*	*CD*	*DE*	*EF*	*FG*	*AG*	*AD*	*AE*	*BG*
1	1	0	0	0	0	0	1	1	1	0
1	1	1	0	0	0	0	0	0	0	1
1	0	1	1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	1	0	0
1	0	0	0	1	1	0	0	0	1	0
1	0	0	0	0	1	1	0	0	0	0
1	0	0	0	0	0	1	1	0	0	1
3	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	0	0	0	1	1	0
2	1	1	1	0	0	0	1	1	0	1
2	1	0	0	0	1	1	1	0	1	1
1	1	0	0	0	0	0	1	0	0	1
2	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	0	0	1	1	0

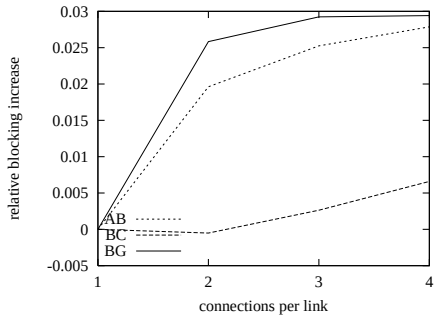
Link balanced traffic

AB	*BC*	*CD*	*DE*	*EF*
0.250	0.500	0.500	0.250	0.500

FG	*AG*	*AD*	*AE*	*BG*
0.500	0.250	0.250	0.250	0.250



Absolute blocking in ESN and WDM networks—link balanced traffic (7starB-lb-blck)

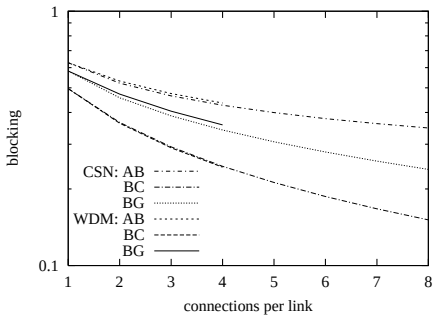


Relative blocking differences between ESN and WDM networks—link balanced traffic (7starB-lb-diff)

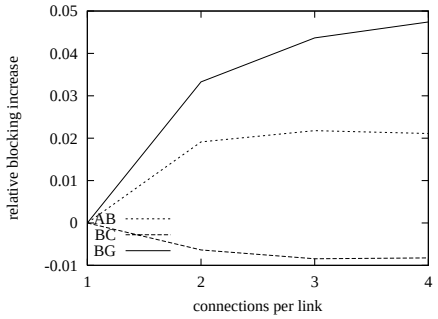
Path uniform & length dependent traffic

AB	*BC*	*CD*	*DE*	*EF*
0.300	0.300	0.300	0.300	0.300

FG	*AG*	*AD*	*AE*	*BG*
0.300	0.300	0.300	0.300	0.300

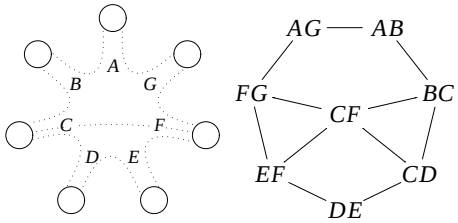


Absolute blocking in ESN and WDM networks—path uniform traffic (7starB-pu-blck)



Relative blocking differences between ESN and WDM networks—path uniform traffic (7starB-pu-diff)

B.3.3 Star network with slightly asymmetric route pattern



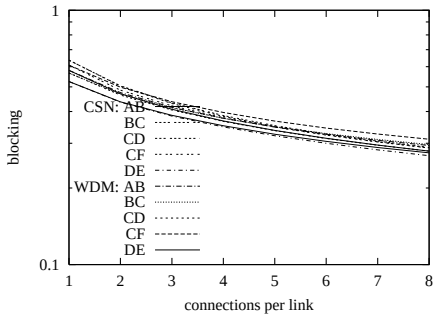
WDM network routes *AB*, *BC*, *CD*, *DE*, *EF*, *FG*, *AG*, *AD*, *AE*, and *BG* with interference graph

c	*AB*	*BC*	*CD*	*DE*	*EF*	*FG*	*AG*	*CF*
1	1	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1
1	0	0	1	1	0	0	0	0
1	0	0	0	1	1	0	0	0
1	0	0	0	0	1	1	0	1
1	0	0	0	0	0	1	1	0
3	1	1	1	1	1	1	1	1
2	1	1	0	0	0	1	1	1

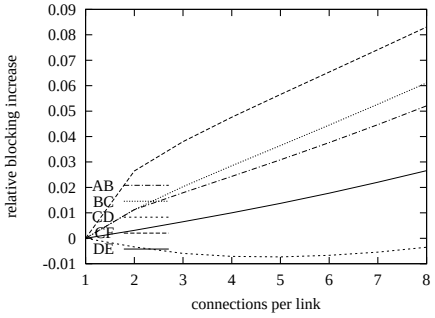
Link balanced traffic

AB	*BC*	*CD*	*DE*
0.250	0.500	0.500	0.250

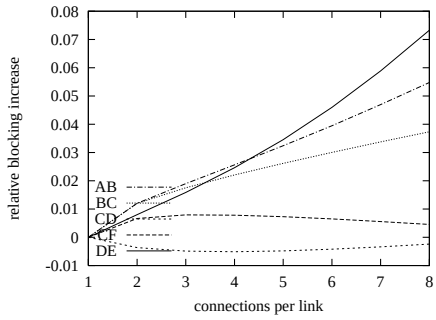
EF	*FG*	*AG*	*CF*
0.500	0.500	0.250	0.250



Absolute blocking in ESN and WDM networks—link balanced traffic
(7starC-lb-blck)



Relative blocking differences between ESN and WDM networks—link balanced traffic (7starC-lb-diff)

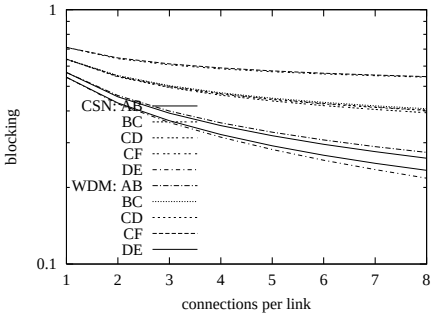


Relative blocking differences between ESN and WDM networks—path uniform traffic (7starC-pu-diff)

Path uniform & length dependent traffic

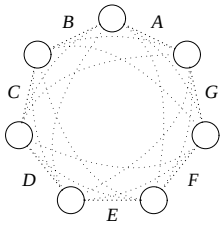
AB	*BC*	*CD*	*DE*
0.500	0.500	0.500	0.500

EF	*FG*	*AG*	*CF*
0.500	0.500	0.500	0.500



Absolute blocking in ESN and WDM networks—path uniform traffic
(7starC-pu-blck)

B.3.4 Ring network with all-to-all routes



WDM network routes *A*, *B*, *C*, *D*, *E*, *F*, *G*, *AB*, *BC*, *CD*, *DE*, *EF*, *FG*, *AG*, *ABC*, *BCD*, *CDE*, *DEF*, *EFG*, *AFG* and *ABG*

The resulting constraint/route table is shown in Figure B.1.

c	A	B	C	D	E	F	G	AB	BC	CD	DE	EF	FG	AG	ABC	BCD	CDE	DEF	EFG	AFG	ABG
1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1
1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1
1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0
1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0
1	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0
1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0
1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	1
3	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	0	1	1
2	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	1	1	1
2	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	1	1	1	1
2	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	0	1
2	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	1	1	1	0
2	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1

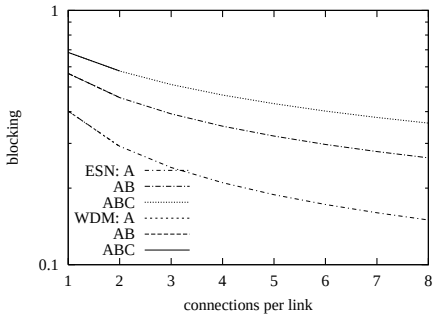
Figure B.2: Constraint / route table for 7 ring network with all-to-all routes

Link balanced & path uniform traffic

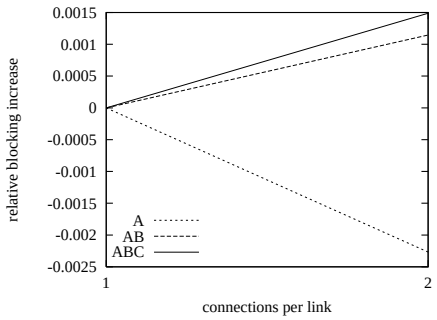
A	B	C	D	E	F	G
0.166	0.166	0.166	0.166	0.166	0.166	0.166

AB	BC	CD	DE	EF	FG	AG
0.166	0.166	0.166	0.166	0.166	0.166	0.166

ABC	BCD	CDE	DEF	EFG	AFG	ABG
0.166	0.166	0.166	0.166	0.166	0.166	0.166



Absolute blocking in ESN and WDM networks—link balanced traffic (7ringA-lb-blck)



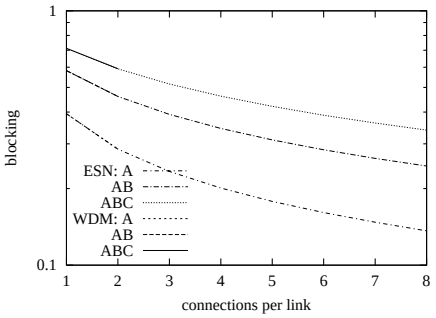
Relative blocking differences between ESN and WDM networks—link balanced traffic (7ringA-lb-diff)

Length dependent

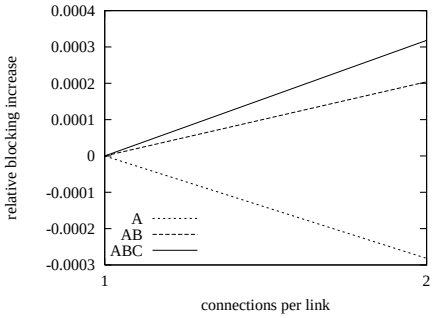
A	*B*	*C*	*D*	*E*	*F*	*G*
0.300	0.300	0.300	0.300	0.300	0.300	0.300

AB	*BC*	*CD*	*DE*	*EF*	*FG*	*AG*
0.150	0.150	0.150	0.150	0.150	0.150	0.150

ABC	*BCD*	*CDE*	*DEF*	*EFG*	*AFG*	*ABG*
0.100	0.100	0.100	0.100	0.100	0.100	0.100

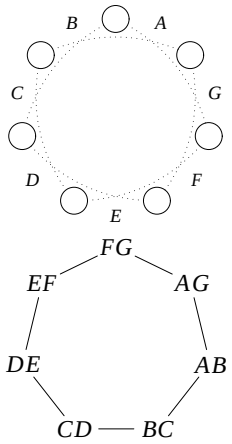


Absolute blocking in ESN and WDM networks—link balanced traffic
(7ringA-ld-blck)



Relative blocking differences between ESN and WDM networks—link balanced traffic (7ringA-ld-diff)

B.3.5 Ring network with routes of length 2

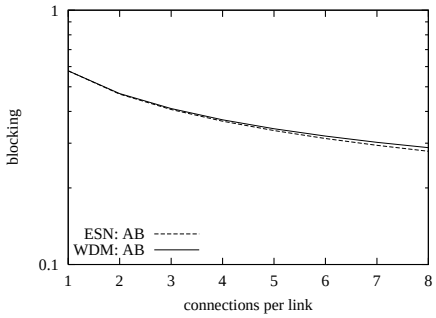


WDM network routes *AB*, *BC*, *CD*, *DE*, *EF*, *FG* and *AG*

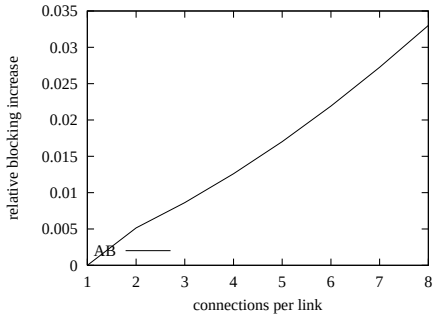
c	*AB*	*BC*	*CD*	*DE*	*EF*	*FG*	*AG*
1	1	0	0	0	0	0	1
1	1	1	0	0	0	0	0
1	0	1	1	0	0	0	0
1	0	0	1	1	0	0	0
1	0	0	0	1	1	0	0
1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	1
3	1	1	1	1	1	1	1

Link balanced, path uniform & length dependent traffic

AB	*BC*	*CD*	*DE*	*EF*	*FG*	*AG*
0.166	0.166	0.166	0.166	0.166	0.166	0.166

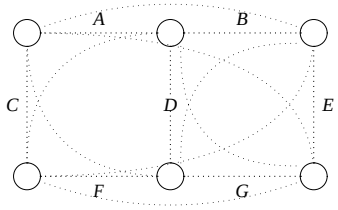


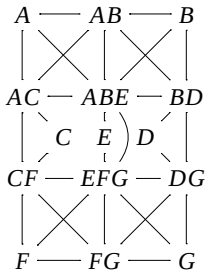
Absolute blocking in ESN and WDM networks—link balanced traffic
(7ringB-lb-blck)



Relative blocking differences between ESN and WDM networks—link balanced traffic (7ringB-lb-diff)

B.3.6 Double-ring network with all-to-all routes



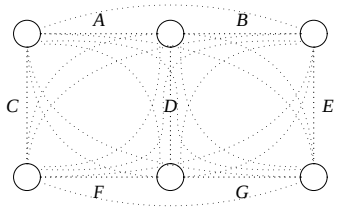


WDM network routes *A*, *B*, *C*, *D*, *E*, *F*, *G*, *AB*, *AC*, *BD*, *CF*, *DG*, *FG*, *ABE* and *EFG*

c	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*AB*	*AC*	*BD*	*CF*	*DG*	*FG*	*ABE*	*EFG*
1	1	0	0	0	0	0	0	1	1	0	0	0	0	1	0
1	0	1	0	0	0	0	0	1	0	1	0	0	0	1	0
1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0
1	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1
1	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1

It is obvious from the constraint/route table that there is no difference in the blocking probabilities of the WDM and ESN networks.

B.3.7 Double-ring network with routes of length 1, 2 & 3



WDM network routes *A*, *B*, *C*, *D*, *E*, *F*, *G*, *AB*, *AC*, *AD*, *BD*, *BE*, *CF*, *DF*, *DG*, *EG*, *FG*, *ABC*, *ABE*, *ADG*, *BDF*, *CFG* and *EFG*

Link balanced traffic

A	*B*	*C*	*D*	*E*	*F*	*G*
0.100	0.100	0.400	0.100	0.400	0.100	0.100

AB	*AC*	*AD*	*BD*	*BE*
0.150	0.150	0.150	0.150	0.150

CF	*DF*	*DG*	*EG*	*FG*
0.150	0.150	0.150	0.150	0.150

ABC	*ABE*	*ADG*	*BDF*	*CFG*	*EFG*
0.150	0.150	0.150	0.150	0.150	0.150

Relative blocking differences between ESN and WDM networks—link balanced traffic (7dringB-lb-diff)

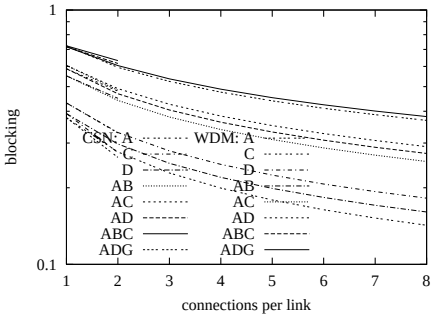
Path uniform traffic

A	*B*	*C*	*D*	*E*	*F*	*G*
0.150	0.150	0.150	0.150	0.150	0.150	0.150

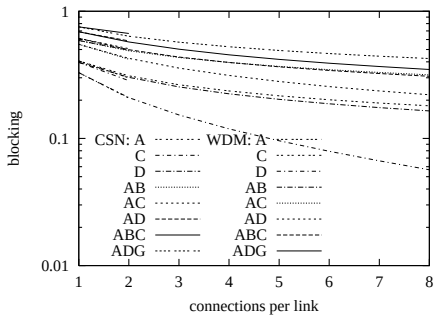
AB	*AC*	*AD*	*BD*	*BE*
0.150	0.150	0.150	0.150	0.150

CF	*DF*	*DG*	*EG*	*FG*
0.150	0.150	0.150	0.150	0.150

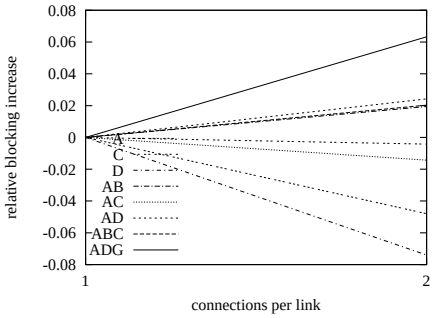
ABC	*ABE*	*ADG*	*BDF*	*CFG*	*EFG*
0.150	0.150	0.150	0.150	0.150	0.150

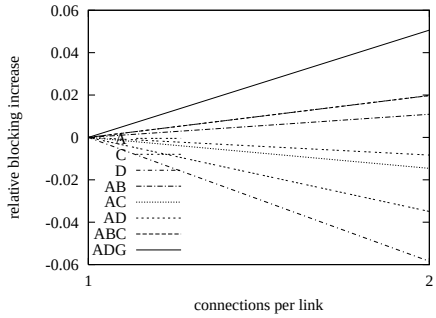


Absolute blocking in ESN and WDM networks—link balanced traffic (7dringB-lb-blck)



Absolute blocking in ESN and WDM networks—link balanced traffic (7dringB-pu-blck)





c	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*AB*	*AC*	*AD*	*BD*	*BE*	*CF*	*DF*	*DG*	*EG*	*FG*	*ABC*	*ABE*	*ADG*	*BDF*	*CFG*	*EFG*
1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	1	0	1	0	0
1	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	1	0	0
1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1
1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	1	1
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1
2	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	1	1	0	1	1
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	0
2	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	1	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1
2	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1	0	0	1	1	1	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0
2	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1	1	0	1	1	1	1	0	1
1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0
3	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	1	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0
2	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1	0
2	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	1	0	1	1	1
2	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	1	1	1	0	1	1
2	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	1

Figure B.3: Constraint / route table for 7 double-ring network with all-to-all routes

Relative blocking differences between ESN and WDM networks—link balanced traffic (7dringB-pu-diff)

Relative blocking differences between ESN and WDM networks—link balanced traffic (7dringB-ld-diff)

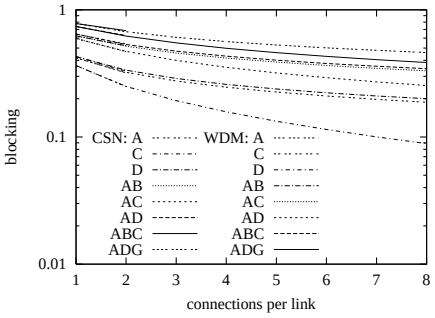
Length dependent traffic

A	*B*	*C*	*D*	*E*	*F*	*G*
0.300	0.300	0.300	0.300	0.300	0.300	0.300

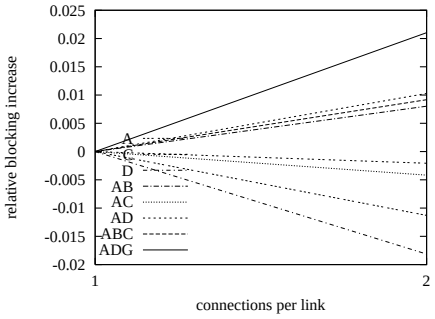
AB	*AC*	*AD*	*BD*	*BE*
0.150	0.150	0.150	0.150	0.150

CF	*DF*	*DG*	*EG*	*FG*
0.150	0.150	0.150	0.150	0.150

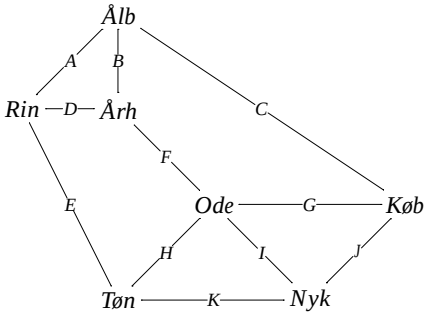
ABC	*ABE*	*ADG*	*BDF*	*CFG*	*EFG*
0.100	0.100	0.100	0.100	0.100	0.100



Absolute blocking in ESN and WDM networks—link balanced traffic (7dringB-ld-blck)



B.4 Danish national network with 11 links



Danish national network.

B.4.1 Gravitational model A

The demand numbers are shown in the upper, and the routes in the lower triangular matrix:

	Ålb	*Rin*	*Årh*	*Ode*	*Køb*	*Tøn*	*Nyk*
Ålb		4	24	16	28	8	12
Rin	*A*		6	4	7	2	3
Årh	*B*	*D*		24	42	12	18
Ode	*BF*	*DF*	*F*		28	8	12
Køb	*C*	*DFG*	*FG*	*G*		14	21
Tøn	*AE*	*E*	*FH*	*H*	*JK*		6
Nyk	*CJ*	*EK*	*FI*	*I*	*J*	*K*	

All demand values are normalised by (i.e. divided by) 112.

c	*A*	*B*	*BF*	*C*	*AE*	*CJ*	*D*	*DF*	*DFG*	*E*	*EK*	*F*	*FG*	*FH*	*FI*	*G*	*H*	*I*	*JK*	*J*	*K*
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

It is obvious from the constraint / route table that there is no difference in the blocking probabilities of the WDM and ESN networks.

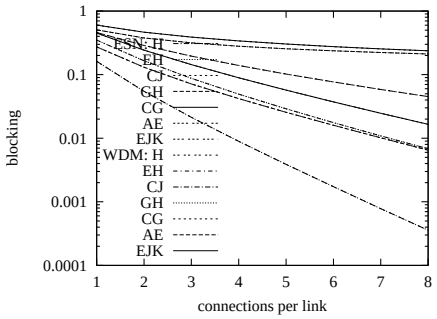
B.4.2 Gravitational model B

The demand numbers are identical, but the routes are different:

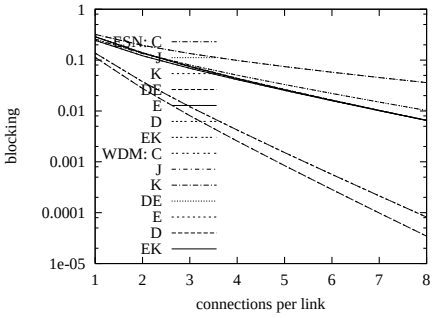
	Ålb	*Rin*	*Årh*	*Ode*	*Køb*	*Tøn*	*Nyk*
Ålb		4	24	16	28	8	12
Rin	*A*		6	4	7	2	3
Årh	*B*	*D*		24	42	12	18
Ode	*CG*	*EH*	*F*		28	8	12
Køb	*C*	*EJK*	*FG*	*G*		14	21
Tøn	*AE*	*E*	*DE*	*H*	*GH*		6
Nyk	*CJ*	*EK*	*FI*	*I*	*J*	*K*	

All demand values are normalised by (i.e. divided by) 100.

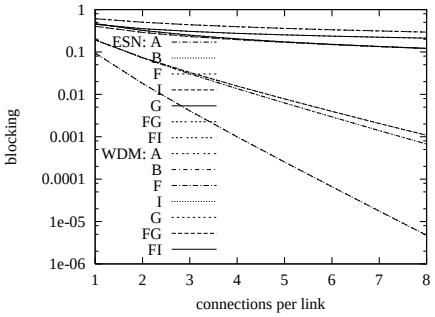
c	*A*	*B*	*BF*	*C*	*AE*	*CJ*	*D*	*DF*	*DFG*	*E*	*EK*	*F*	*FG*	*FH*	*FI*	*G*	*H*	*I*	*JK*	*J*	*K*
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1



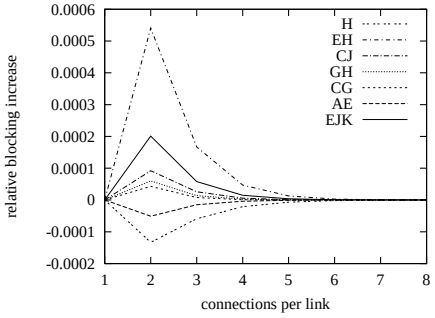
Absolute blocking in ESN and WDM networks—gravitational traffic
(DenmarkB-gm-blck-1)



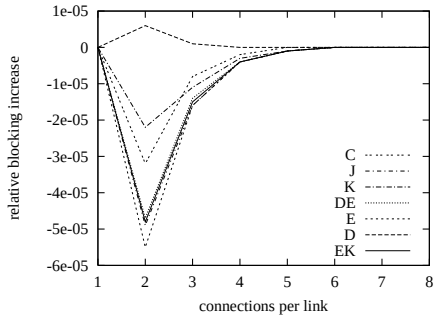
Absolute blocking in ESN and WDM networks—gravitational traffic
(DenmarkB-gm-blck-2)



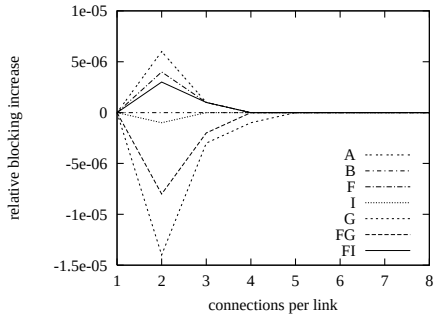
Absolute blocking in ESN and WDM networks—gravitational traffic
(DenmarkB-gm-blck-3)



Relative blocking differences between ESN and WDM networks—gravitational traffic
(DenmarkB-gm-diff-1)



Relative blocking differences between ESN and WDM networks—gravitational traffic
(DenmarkB-gm-diff-2)



Relative blocking differences between ESN and WDM networks—gravitational traffic
(DenmarkB-gm-diff-3)

Appendix C

Internet dial-up teletraffic

This appendix describes a study of arrival processes at internet dialup modems located in local exchanges.

C.1 Introduction

Until the asymmetric digital subscriber line (ADSL) technology was introduced widely in Denmark at the turn of the millennium, the main mode of connection for residential and small-business users was via modem dial-up. The majority of users utilised analog modems running over public switched telephony system (PSTN) protocols, but also integrated services digital network (ISDN) modems (or protocol converters) were being used. The (narrowband) ISDN protocol provides two digital (bearer) channels, and a signalling (data) channel, which in Denmark is not available for user data streams. Apart from giving access to the full 64kbit/s channel bandwidth, ISDN connections have the added advantage of allowing bundling, i.e. using both lines simultaneously for 128kbit/s bandwidth, or concurrent data and voice access, which makes them tractable especially for small businesses.

Although ADSL to a great extent is taking over this data market, analog and ISDN modems will still continue to exist for several years to come, and characteristics of dial-up Internet calls are still of interest when dimensioning the local access network (Färber et al., 1999; Møller and Rasmussen, 1998; Naldi, 1999). Recent research suggests that Internet calls have characteristics which are different from ordinary voice calls; the average holding times are longer, which among other things impacts the observed call blocking (Fredericks, 1999).

C.1.1 Discrete data recordings

This section deals with analysis and modelling of real Internet dial-up traffic based on extensive data, which are described and analysed. One key observation is that when recording the traffic process in modern digital switches, the time axis is digitalised and the analysis and modelling must take this into account. If the length of the discrete time intervals are comparable with the holding times, or if there are zero-length holding times, the effect is significant, and analysing the observed data without considering the recording method can lead us to the wrong conclusions regarding the nature of the arrival process.

Another important point is to distinguish between fast variations within short time intervals and slow variations during the day. If we do so, we are able to characterise the traffic in a predictable way which allows sensible capacity planning that provides acceptable service without wasting equipment bandwidth.

The detailed analysis of the data presented here shows that the arrival process can be modeled as a Poisson process with slow intensity variations.

C.1.2 Field data

The field data used for the study are Internet dial-up calls supplied by Tele Danmark. During one week, Monday 18 – Sunday 24, January 1999, about 3.8 million calls have been recorded. Every call is recorded with arrival time, holding time, and other useful information such as identity of modem-pool, ISDN / PSTN, subscription type, etc. Only about 0.5% of the call attempts are rejected for various reasons, so we can assume that the data represents the offered traffic. About 79% of all calls are PSTN, and the remaining 21% are ISDN calls. ISDN calls are always counted as one call, independently of the number of channels occupied, so the process can be considered regular. The accuracy of the time recordings is one second.

The daily traffic profile has the maximum in the evening; Figure C.1 shows a typical profile for the number of calls in 15 minute intervals. Similar patterns with two peaks just before and after dinner time, and a valley in the nighttime have been found by other authors (Møller and Rasmussen, 1998; Naldi, 1998; Blair, 1998).

The mean value of the holding times is 5–10 times bigger than the mean value of voice holding times, and a typical pattern is shown in Figure C.2. The distribution of the holding time has a coefficient of variation which is bigger than for voice calls, and the modelling has been studied using hyper-exponential, log-normal and Pareto distributions (Färber et al., 1999).

In the following we will focus upon the arrival process; Møller and Rasmussen (1998) concluded that the arrival process of Internet calls is more bursty than the

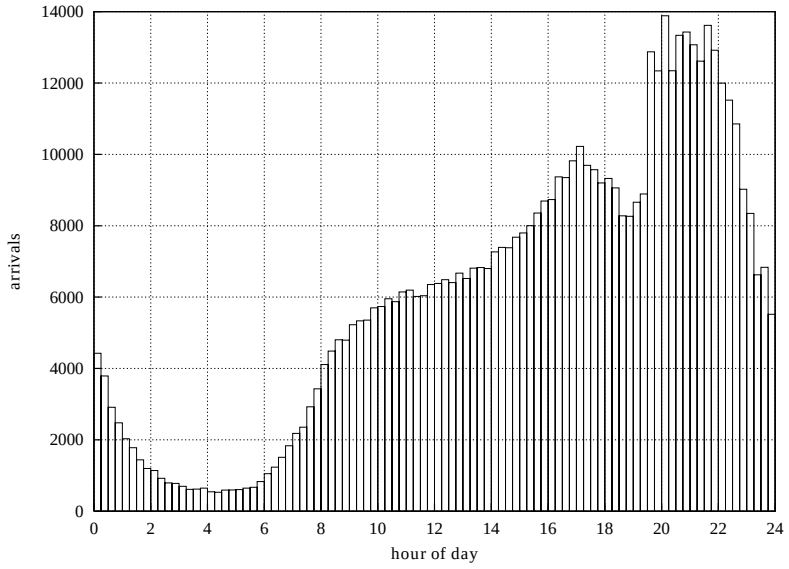


Figure C.1: Number of calls per 15 minutes, Tuesday 19.01.1999.

Poisson process, whereas Naldi (1999) concludes that within intervals of up to 15 minutes it can be considered as a stationary Poisson process. In larger time intervals we have intensity variations, but still a Poisson process.

This corresponds to voice traffic where we have slow variations due to the daily call profile, but where the fast variations are of random (Poissonian) nature. The concept of slow and fast variations was introduced by Palm (1987) and studied in detail for voice traffic by Iversen (1973).

C.1.3 Studying arrival processes

There are two basic ways to study an arrival process: We may keep a time interval fixed and study the number of calls within this interval (*number representation*), or we may keep a number of calls fixed, and study the time interval to get this number of calls (*interval representation*).

When the continuous real time is discretized by the modern digital switch to an integer, the *interval representation* encounters additional problems, whereas the *number representation* fits with this approach. We consider the two representations in the following sections.

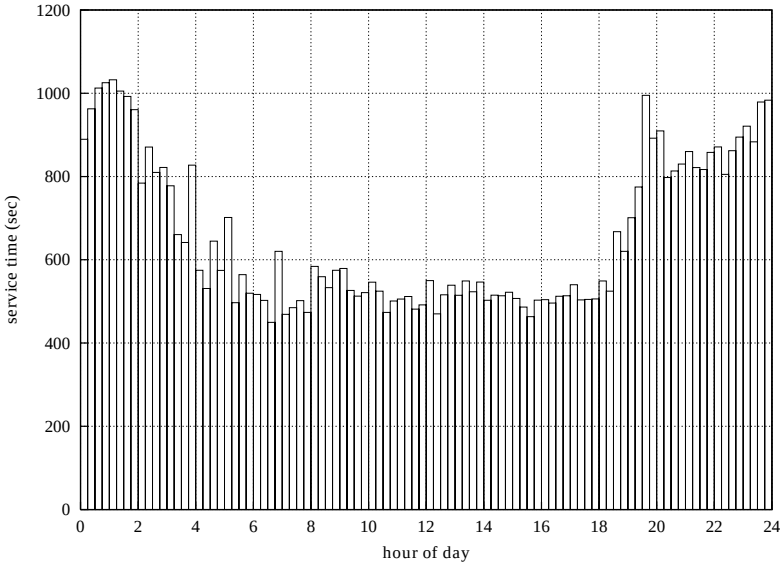


Figure C.2: Mean holding time in seconds for calls arriving during 15 minutes, Tuesday, 19.01.1999.

C.2 Interval representation

When observing a continuous time process at regular time instants the continuous time intervals are transformed into a discrete time distribution. The important case of the Poisson process is transformed into a discrete time process with correlation between inter-arrival times of duration zero. Thus a model in discrete time based on geometric inter-arrival times (Bernoulli arrival process) is *not* equivalent to a Poisson process in continuous time.

The transformation of traffic processes from continuous time to discrete time has been dealt with in the theory of traffic measurements where we observe the traffic process at regular time intervals (Iversen, 1973; Westerberg, 1975). The discrete time increases the observed burstiness of the traffic process, so it looks heavy-tailed, even if it is really Poissonian.

C.2.1 The scanning method

In general one can consider scanning intervals with a given distribution, but in the following we will only consider regular (constant-length) scanning intervals. The scanning method is e.g. applied to traffic measurements, call charging, numerical simulations, and processor control. By the scanning method we observe a discrete time distribution for the holding time which in real time usually is continuous. At each scanning time the accumulated effect of the previous interval is observed, and this will round each real time interval up or down, cf. Figure C.3. An example

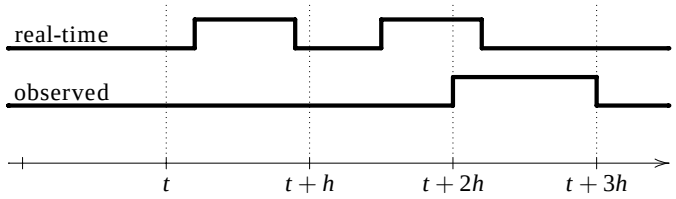


Figure C.3: The scanning method will round each real-time interval up or down

of how the observed arrival counts—the original empirical data—are converted into interarrival times is shown in Figure C.4. It can be seen that the higher the arrival intensities are, the more zero-interarrival times will be computed.

Choosing the constant distance h between scanning instants, we are able to derive a relation between the observed time interval and the real time interval, cf. Figure C.5. Note that there are overlaps between the continuous time intervals, so that the discrete distribution cannot be obtained by a simple integration of the

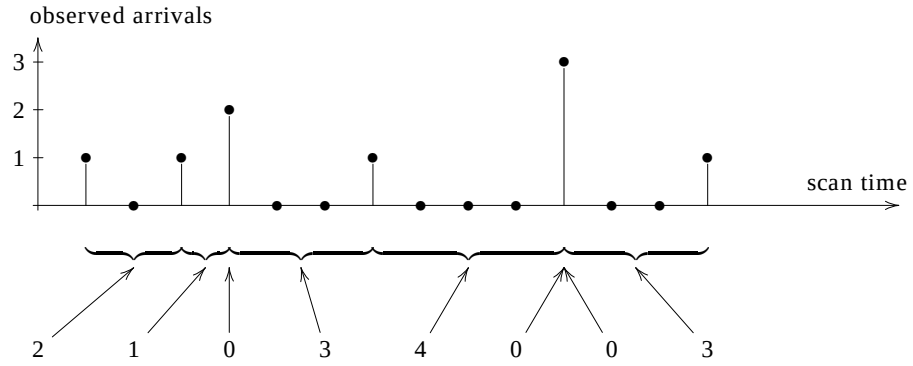


Figure C.4: Extracting interarrival times from observed arrival counts

continuous time interval over a fixed interval of length h. If the real holding times have a distribution function $F(t)$, then it can be shown that we will observe the following discrete distribution (Iversen, 1973):

$$p(0) = \frac{1}{h}\int_0^h F(t)\,dt \tag{C.1}$$

$$p(k) = \frac{1}{h}\int_0^h \{F(t+kh) - F(t+(k-1)h)\}dt, \quad k = 1,2,\ldots \tag{C.2}$$

Interpretation: The arrival time of the call is assumed to be independent of the scanning process. Therefore, the density function of the time interval from the call arrival instant to the first scanning time is uniformly distributed and equal to $1/h$. The probability of observing zero scanning instants during the call holding time is denoted by $p(0)$ and is equal to the probability that the call terminates before the next scanning time. For a fixed value of the holding time t this probability is equal to $(F(t)/h)$, and to obtain the total probability we integrate over all possible values t $(0 \leq t < h)$ and get (C.1). In a similar way we derive formula (C.2) for $p(k)$.

For exponentially distributed holding time intervals, $F(t) = 1 - e^{-\mu t}$, we will observe a discrete distribution, *Westerberg's distribution* (Westerberg, 1975; Iversen, 1973):

$$p(0) = 1 - \frac{1}{\mu h}\left(1 - e^{-\mu h}\right) \tag{C.3}$$

$$p(k) = \frac{1}{\mu h}\left(1 - e^{-\mu h}\right)^2 \cdot e^{-(k-1)\mu h}, \quad k = 1,2,\ldots \tag{C.4}$$

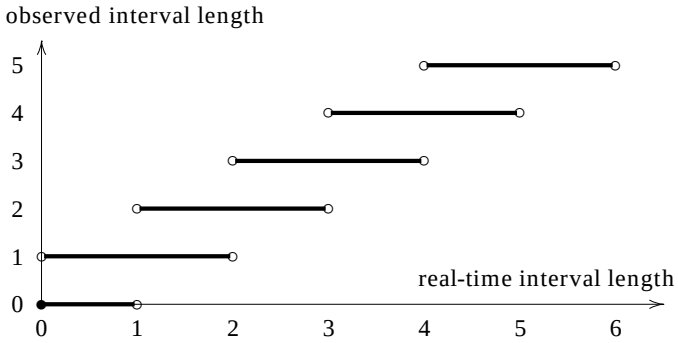


Figure C.5: By the scanning method a continuous time interval is transformed into a discrete time interval. The transformation is not unique.

The i'th derivative of the probability generating function $\mathcal{Z}(z)$ (Z–transformed) for the value $z = 1$ becomes:

$$\mathcal{Z}^{(i)}(1) = \frac{i!}{i\,h} \cdot \frac{1}{\left(e^{\mu h} - 1\right)^{i-1}} \tag{C.5}$$

from which we find the mean value m_1 and form factor ε (second moment divided by the squared mean value) (Iversen, 1998):

$$m_1 = \mathcal{Z}^{(1)}(1) = \frac{1}{\mu\,h}\,, \tag{C.6}$$

$$\sigma^2 = \mathcal{Z}^{(2)}(1) + \mathcal{Z}^{(1)}(1) - \mathcal{Z}^{(1)}(1)^2\,, \tag{C.7}$$

$$\varepsilon = \left(\frac{\sigma}{m_1}\right)^2 = \mu\,h \cdot \frac{e^{\mu h} + 1}{e^{\mu h} - 1} \geq 2\,. \tag{C.8}$$

Thus we will *always observe the correct mean value;* this can be shown to be true for any distribution function $F(t)$. When using Karlsson charging we will therefore always in the long run charge the "correct amount."

For a continuous measurement the form factor is 2. The contribution $\varepsilon - 2$ is thus due to the influence from the measuring principle. Christensen et al. (2000) generalise the scanning method to phase-type distributions for both the scanning interval and the inter-arrival time.

The form factor is a measure of accuracy of the measurements. Figure C.6 shows how the form factor of the observed holding time for exponentially distributed holding times depends on the length of the scanning interval (C.8). By

continuous measurements we get an ordinary sample. By the scanning method we get a sample of a sample so that there is uncertainty both due to the measuring method and due to the limited sample size.

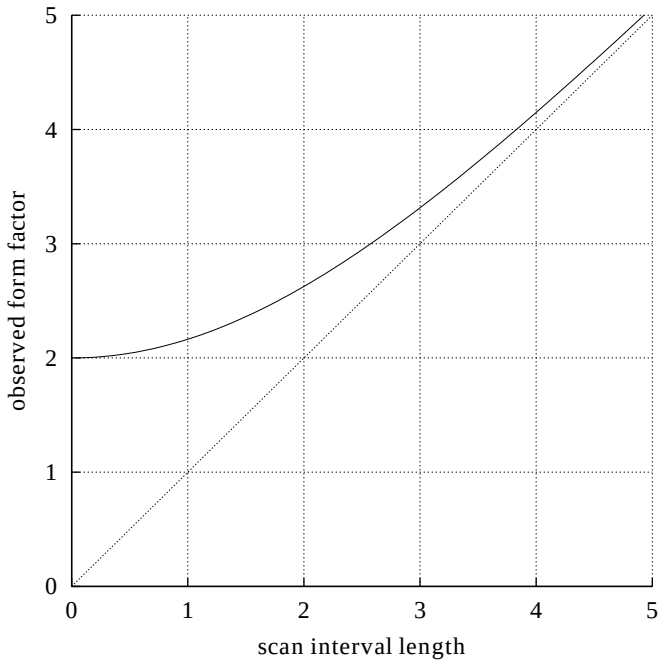


Figure C.6: *Form factor for exponentially distributed holding times which are observed in constant scanning intervals in an unlimited measuring period. A scan interval length $h = 0$ corresponds to a continuous measurement. Note that we loose almost no information (the form factor is close to 2) if the scan interval is smaller than the mean holding time (chosen as time unit).*

In Figure C.7 and Figure C.8 we compare for ISDN calls the observed formfactor with the formfactor calculated from Westerberg's distribution using the total number of calls during 15 minutes to calculate the mean arrival rate. We notice the two curves are very similar—the ratio between the two results is shown in Figure C.9—so even though the formfactor is greater than two, we may assume it is a Poisson process within the 15 minute intervals.

If we look at high arrival rates (for instance by considering all calls, not only ISDN or single modem pools), the observed and calculated form factors will be

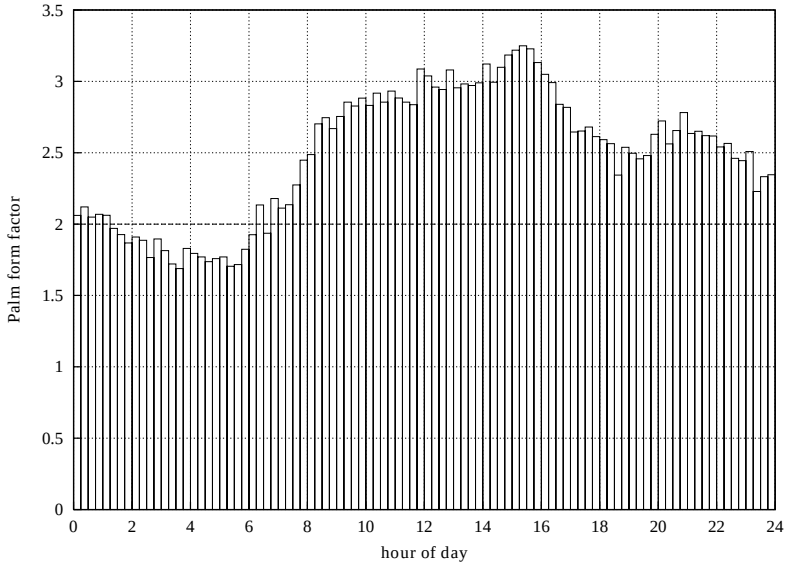


Figure C.7: Observed formfactor of inter-arrival times for ISDN-calls in 15-minutes intervals from observed data, Tuesday 19.01.1999. Cf. Figure C.8.

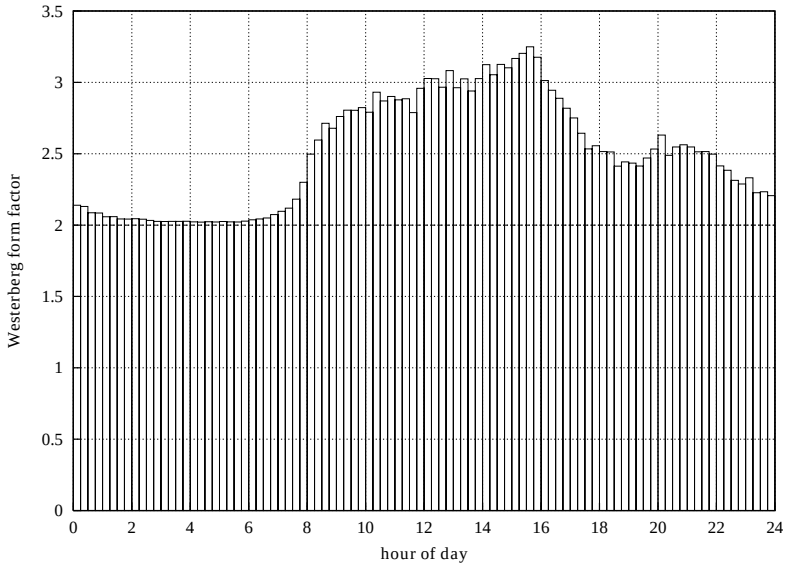


Figure C.8: Formfactor of inter-arrival times for ISDN-calls calculated from Westerberg's distribution (C.8) based on the total number of calls observed per 15 minutes, Tuesday 19.01.1999. Cf. Figure C.7.

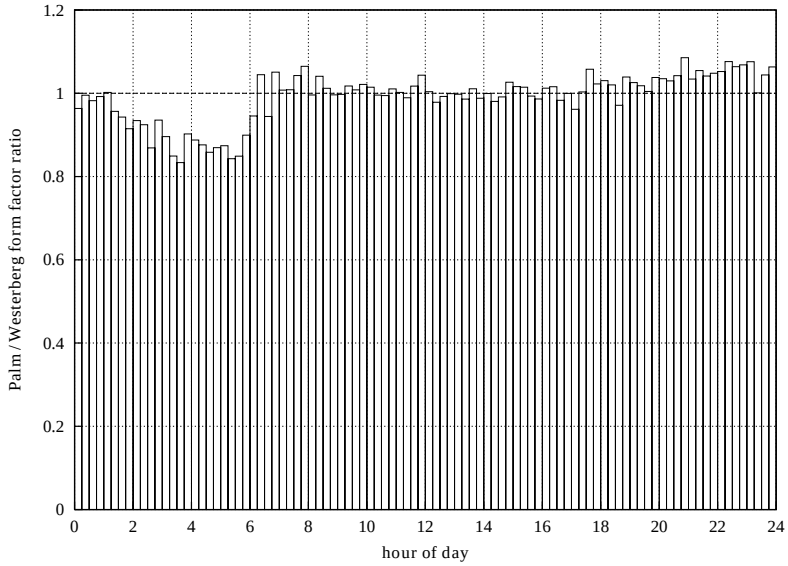


Figure C.9: The ratio between the observed values (Figure C.7) and the theoretical values (Figure C.8).

identical, because all inter-arrival times will be zero except for 900 inter-arrivals equal to one second due to 900 scans during 15 minutes. An example is shown in Figures C.10–C.12. In this case we cannot make any conclusions about the inter-arrival time distribution for the busy periods.

Investigation of intervals with fewer calls, e.g. for individual modem pools or during the night, show that Westerberg's distribution is a very good model for inter-arrival times for periods up to one hour. If we include the ISDN-calls as two simultaneous arrivals, the observed and calculated values become different for the cases where we do not have too many calls per scan interval.

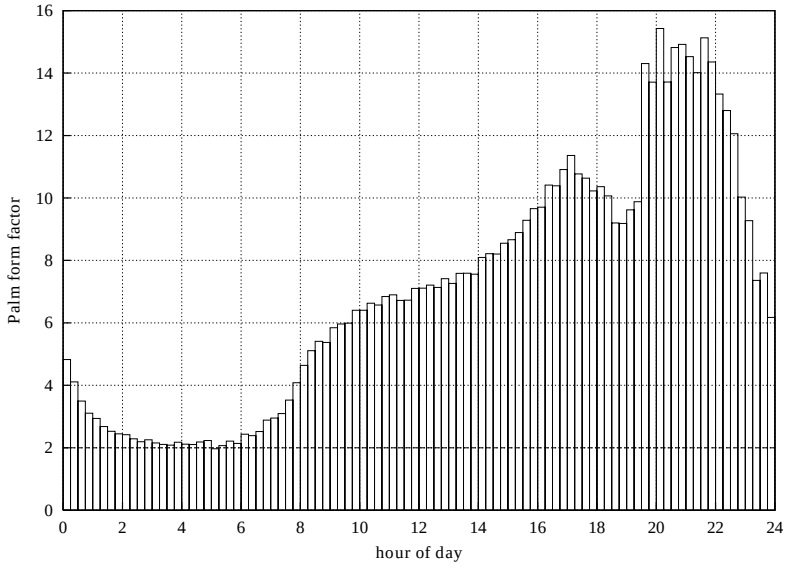


Figure C.10: Observed formfactor of inter-arrival times for all calls in 15-minutes intervals from measurements. Tuesday 19.01.1999.

C.3 Number representation

In this case we study the number of calls during intervals of fixed duration. The measuring method is exact. If the arrival process is a stationary Poisson process, then the number of calls during a fixed time interval will be Poisson-distributed. We may observe the number of calls per time interval (e.g. one second) during 15 minutes and perform tests for the Poisson distribution. In general, a χ^2-test

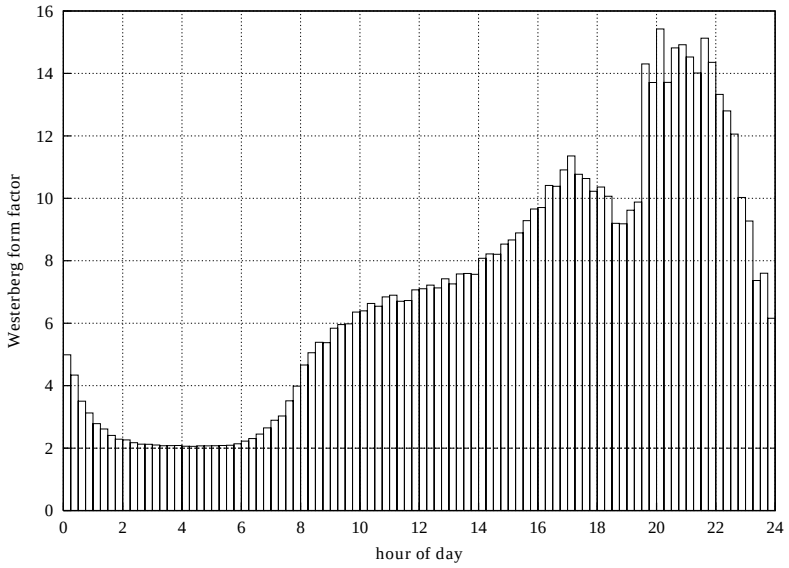


Figure C.11: Formfactor of inter-arrival times for all calls in 15-minutes intervals calculated from Westerberg's distribution based on the total number of calls observed. Tuesday 19.01.1999.

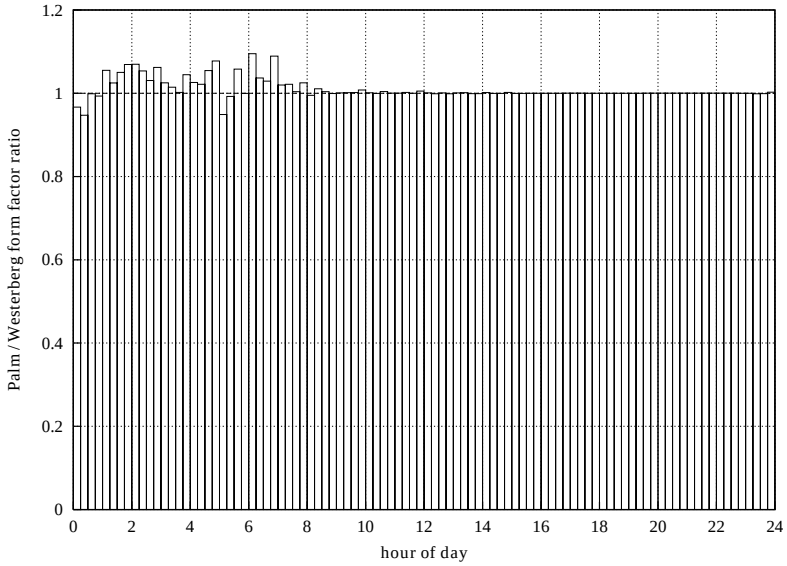


Figure C.12: The ratio between the observed form factors (Figure C.10) and theoretical values (Figure C.11).

will accept a Poisson hypothesis. In the following we shall look at a second order statistics to study intensity variations.

C.3.1 Generalised mean differences

Let us consider a discrete distribution function $p(i)$ ($i = 0, 1, \ldots$) with mean value m_1, second moment m_2, and variance $\sigma^2 = m_2 - m_1^2$.

The generalised mean difference of order r is defined by:

$$\delta_r = \sum_i \sum_{j \neq i} \mid i - j \mid^r \; p(i)\, p(j)\,.$$

For $r = 2$ we have

$$\begin{aligned}
\delta_2 &= \sum_i \sum_{j \neq i} (i-j)^2 \, p(i)\, p(j) \\
&= \sum_i \sum_j (i-j)^2 \, p(i) p(j) \\
&= \sum_i \sum_j \left\{ i^2 \, p(i)\, p(j) + j^2 \, p(j)\, p(i) - 2\, i\, j \, p(i)\, p(j) \right\} \\
&= \sum_i i^2 \, p(i) + \sum_j j^2 \, p(j) - 2 \left(\sum_i i \, p(i) \right) \left(\sum_j j \, p(j) \right) \\
&= 2 \left(m_2 - m_1^2 \right) \\
&= 2\sigma^2
\end{aligned}$$

For the discrete distributions applied in the BPP (Binomial–Poisson–Pascal) traffic model we get the normalised generalised mean difference (NGMD) (In the Binomial and Pascal distributions p denotes the probability of success):

$$\begin{aligned}
&\text{Binomial:} && \tfrac{\delta_2}{m_1} = 2(1-p) \\
&\text{Poisson:} && \tfrac{\delta_2}{m_1} = 2 \\
&\text{Pascal:} && \tfrac{\delta_2}{m_1} = 2(1+p)
\end{aligned}$$

A double stochastic Poisson process (a Cox process) is a Poisson process where the intensity λ is a stochastic variable. If for example λ is Γ-distributed, then the interarrival times are Pareto-distributed (Iversen, 1998). Let λ have the mean

value be m_λ and the variance σ_λ^2, then we get for the the number of calls in a fixed time interval:

Palm: $$\frac{\delta_2}{m_1} = 2 + 2 \cdot \frac{\sigma_\lambda^2}{m_\lambda} .$$

The above statistics may be used to distinguish between fast and slow variations in a point process (Iversen, 1973). If we let $p(i)$ and $p(j)$ denote the number of calls in two consecutive intervals of fixed length, then the slow variations have no influence upon the the result, because the statistic is a linear function of the intensity. On the other hand, fast variations within the duration of the two intervals will increase the value of the statistics.

In Figure C.13 we show the NGMD for periods of 15 minutes based on consecutive intervals of duration 1 second each. The average is about 2.2 indicating that the arrival process is a little more bursty than the Poisson process. In Figure C.14 we show the NGMD for the same period when the consecutive intervals are of length 60 seconds each. The mean value increases to 2.53 indicating that there are small intensity variations during 2 minutes.

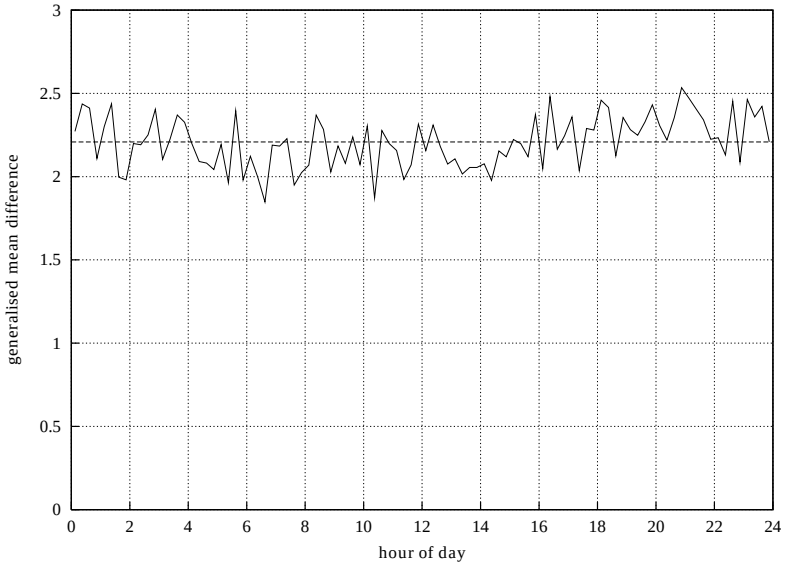


Figure C.13: Normalised generalised mean differences using 1-second intervals for periods of 15 minutes. Average = 2.2079. Tuesday 19.01.1999.

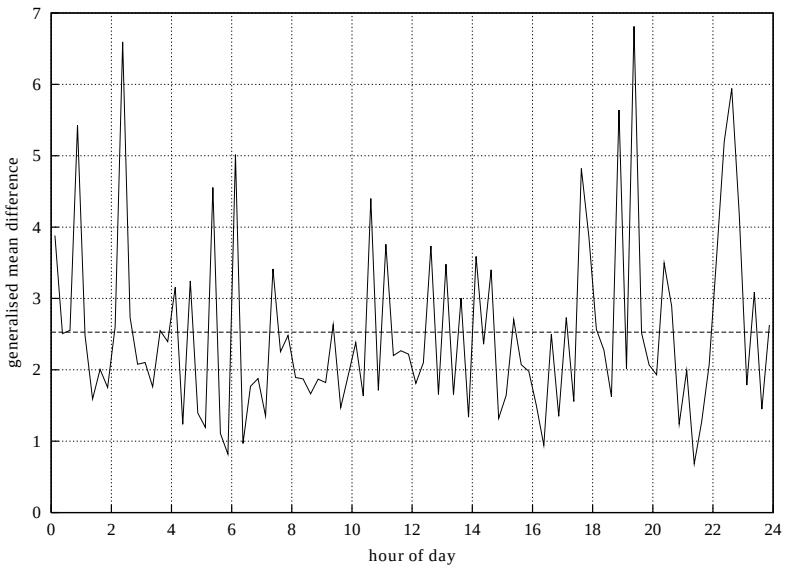


Figure C.14: Normalised generalised mean differences using 60-seconds intervals for periods of 15 minutes. Average = 2.5280. Tuesday 19.01.1999.

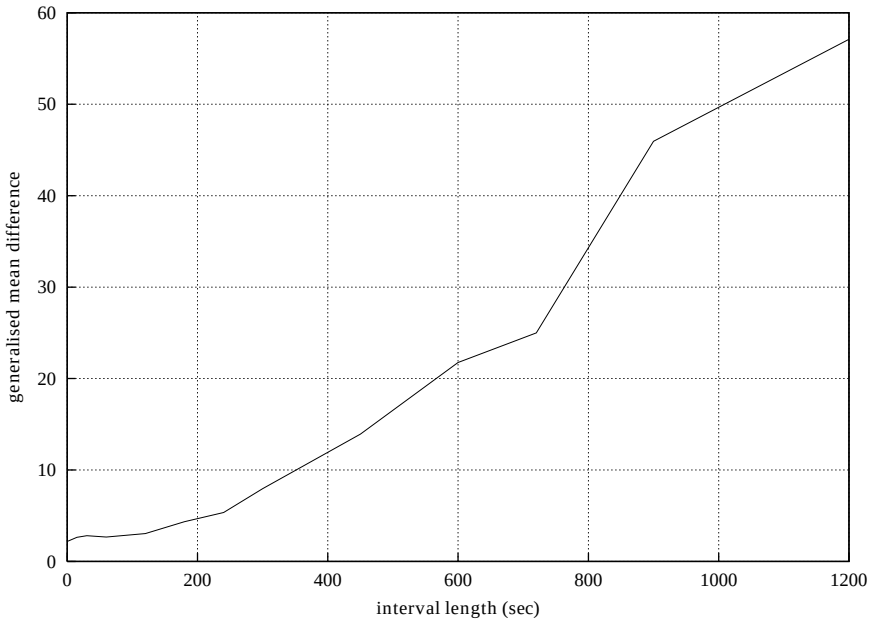


Figure C.15: The mean-squared-deviation for all calls as a function of the period. Tuesday 19.01.1999.

In Figure C.15 we show the NGMD for increasing values of the consecutive intervals from 1 to 1200 seconds. The curve indicates significant intensity variations during two consecutive periods of each 20 minutes.

If we consider individual model pools, the NGMD becomes close to 2 for intervals of length 5–10 minutes, which indicates a Poisson process. By Palm's theorem (Iversen, 1998) the total arrival process should be even more Poissonian. In the above examples the average number of calls per second is about 7 for all 24 hours of the day. It seems as if the large number of calls per time interval tends to increase the value of NGMD above 2, even for a Poisson process.

C.4 Conclusion

Above we have presented results for the analysis of the arrival process of dial-up Internet calls. Most results are based on all recorded calls. Hussain (1999) has also carried out investigations for individual modem-pools.

All investigations indicate that the arrival process is a Poisson process with slow variations during the day. Only at particular times, e.g. at 19:30 in the evening when the tariff is reduced to 50%, fast variations are observed (cf. Figures C.1 and C.14).